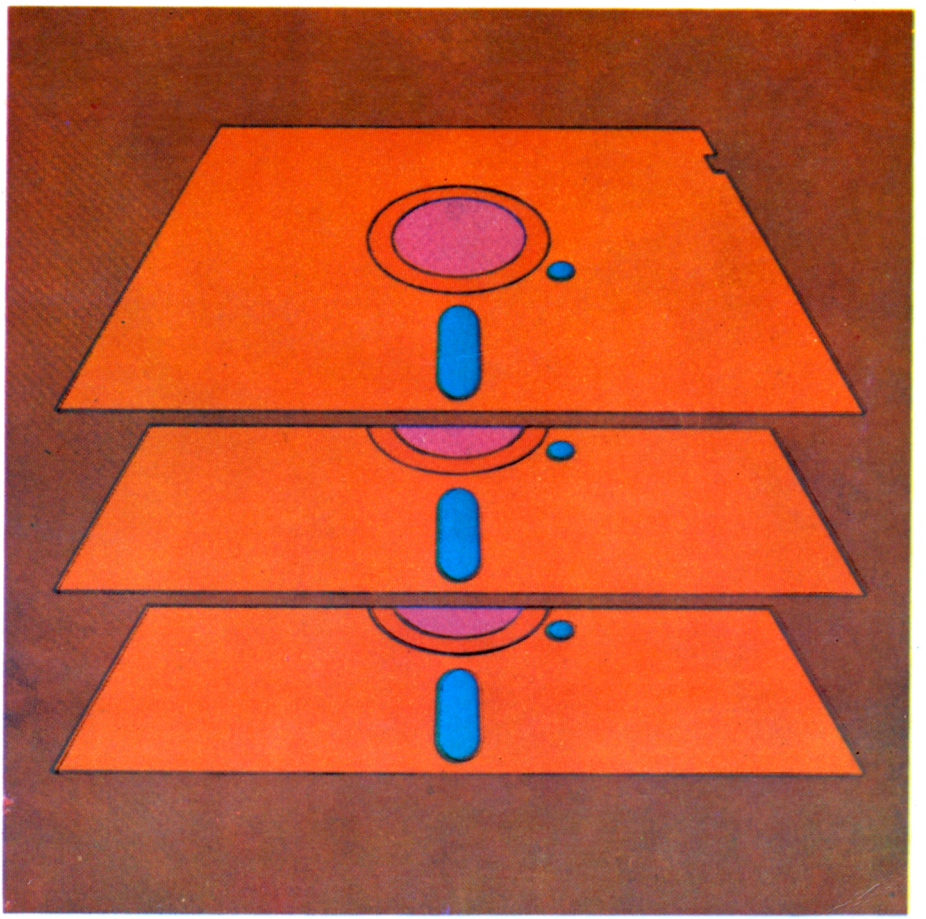
Apple II

APPLE DOS

作業系統使用手冊

蔡毓琛・魏易休 譯



香港科技出版社

APPLE DOS

作業系統使用手冊

蔡毓琛・魏易休 譯

香港科技出版社

APPLE DOS 作業系統使用手册

編譯者：蔡毓琛 • 魏易休
出版者：香港科技出版社
發行者：香港科技出版社
九龍彩虹道810號六樓
印刷者：永達印刷公司
香港黃竹坑建明工業大廈[illegible]

羅拔書屋
$22.00

定　價：H. K.

APPLE DOS作業系統使用手册名詞對照表

（弧號內之譯名亦爲其它手册所使用）

英　　文	譯　　名
Bit	比次（位元）
Boot	未翻或啓動
Booting Loader	靴帶式輸入程式 （逐段載入程式）
Byte	拜（位元組）
Carriage return	回車
Character	字元
Copy	複製
Drive	磁碟機
Initialize	起始或起始設定
Sector	段落（磁扇）
Slave Disk	奴磁碟
Slot	溝槽
Syntax error	語法錯誤
Text file	主文檔（正文檔，文字檔）
Track	軌道（磁軌）

前　言

本手册首要目標是教導使用者如何運用 DOS ，在每章中含有許多程式例子以輔助了解各個 DOS 命令工作情形。本手册的附錄亦可做爲 DOS 的快速參考資料。

APPLE Ⅱ至少須16K 記憶體才能使用磁碟，但因 16K 系統空間過小，最好系統記憶體不小於 32K ，以存放使用者程式。系統主磁碟的MUFFIN 及 FID 程式須要在 32K 或以上的系統執行。存於磁帶或磁碟的APPLESOFT與 DISK Ⅱ同時使用時，系統也至少須 32K 記憶體。

APPLE DISK Ⅱ爲“ floppy disk ”用來快速存取資料。本磁碟爲表面鍍上磁性物質的塑膠碟，直徑約 5 英吋，包裝在一方型硬膠殼中。

APPLE Ⅱ磁碟的使用，最大方便處就是可將資料以檔案名稱檔案化，存取於磁碟上。如：編上“電話號碼”的程式可用下式將它保存於磁碟，亦可用類似簡單命令再取出使用。

SAVE PHONE NUMBERS

其中PHONE NUMBERS爲程式的檔案名稱。

DOS 爲磁碟作業系統的簡稱（ Disk Operating System）。DOS 包括了許多如自動保持檔案軌道，保存及取出資訊，及處理多重管家工作（ Housekeeping ）的程式。有人亦稱 DOS 爲“ doss ”或“ dee oh ess ”。學習 DOS 主要在於學其 DOS 命令。這些命令用來擴展APPLESOFT，INTEGER BASIC，或機器語言程式。

本書中有兩符號常出現：

在段落前，表示此處須注意某不尋常的情況。

在段落前，表示在某狀況之下造成BASIC程式無法回頭（recover），而失去程式。可能須重啓動（restart）DOS或BASIC。

* * * 注 意 * * *

本手册用於 16 段磁碟 DOS 版本的使用（即DOS 3.3 及其後的版本）。若使用者原有早於DOS 3.3 的13段磁碟 DOS 版本，可以先依附錄H換掉磁碟控制卡的 PROMs ，再用MUFFIN 程式將資料移到 16 段的磁碟上，以更新 13 段的 DOS 成爲 16 段的 DOS 。在 16 段的系統上BOOT 13 段磁碟，則可利用附帶的“ BASICS ”磁碟。

在許多情況下，以MUFFIN 轉變的磁碟可能不能工作。這發生在磁碟程式試圖在磁碟本身的特殊位置找指令時。絕大多數程式不屬此類。MUFFIN 一磁碟並不傷害原 13 段的磁碟。13 段磁碟的 BOOT ，須使用到“ BASICS ”程式。

目　錄

第一章　組立與操作　1

機件分解　2
電纜線連接　2
控制板組立　4
多磁碟機系統的組立　6
二號磁碟機（DISK II）及磁碟的保養　6
插入及拉出磁碟　8

第二章　操作開始　11

背景　12
特殊鍵　13
把 DOS 程式從磁碟搬入主記憶體　14
如果 BOOT 動作失靈的話　16
新磁碟的啓始　16
用DOS 系統來載入及保存　19
登錄（CATALOG）　21
在一檔名稱內包含了什麼　22
重新命名檔案的方法　23
刪除檔案的方法　23
不小心重置機器時的修復法　24

第三章 練習使用選擇權 27

磁碟機、溝槽及磁碟册的選擇 28
語法 31
啓始命令（INIT） 31
載入、執行及保存命令（LOAD，RUN及SAVE） 32
刪除命令（DELETE） 34
命令綜觀：BOOT，CATALOG，SAVE，RUN 及 DELETE 35
語言間的轉換：FP及INT命令 36
在程式中如何使用 DOS 命令 38

第四章 安全的操作 43

建出一"轉動鎖"（TURNKEY）系統 44
鎖制及解鎖（LOCK及UNLOCK） 45
查證命令（VERIFY） 46
磁碟的禁寫入 48
防止你遭到不幸的方法 49
複製（COPY）程式使用法 50

第五章 更多的〝管家〞方法 55

除錯：監視及非監視命令（MON及NOMON） 56
MAXFILES 命令 57
追蹤（TRACE）命令 59
如何使用MASTER CREATE程式 60

第六章　如何使用循序主文檔 63

主文檔：簡介 64
循序主文檔：一些例子 66
開啓及關閉循序檔 78
循序檔的寫入 80
循序檔的讀出 86
更多有關循序檔資料：APPEND 及 POSITION 命令
拜定位後處理資料的方法 89

第七章　自動化的 APPLE 97

利用一個主文檔控制APPLE: EXEC命令 98
新建一個 EXEC 檔 100
在主文檔中截取程式 101
將機器語言常式轉換爲BASIC語言 103
執行的過程 104

第八章　隨機存取檔的用法 107

隨機存取檔：如何工作 108
一個特定的資料錄 108
多重資料錄 110
一個隨機存取檔的範例：RANDOM 程式 113
對隨機存取檔的寫入與讀取 115

第九章　機器語言檔的用法 119

機器語言檔 120

BSAVE 120
BLOAD 121
BRUN 122
RWTS 副常式 123

第十章 輸入、輸出、及鏈結 129

選擇輸入／輸出裝置：IN#，PR# 及其他 130
Integer BASIC 鏈結 137
Applesoft 鏈結 138

附錄A DOS命令中用到的幾種檔案型式 141

檔案程式使用情形（按命令字母順序排列） 142
檔案程式使用情形（按檔案型式排列） 143

附錄B DOS訊息 145

ONERR GOTO 碼 147
討論 147

附錄C 磁碟資料的格式 161

儲存過程的簡要說明 162
寫入循序主文檔 163
寫入隨機存取主文檔 164
DOS 如何寫入主文檔：一般過程 165
檔案段落的內容 166
T／S表（軌道／段落表） 167
磁碟目錄表 169

VTOC （磁碟內容表） 171
軌道比次對映 173
軌道和段落的配置順序 175
從磁碟取出資料 177
從循序檔讀取 177
從隨機存取檔讀取 178

附錄D　記憶體之運用 181

BOOT 磁碟時寫入記憶體的區域 182
DOS 及兩種BASIC所使用記憶體的區域 183
BOOT DOS 時所設定的HIMEM值 184

附錄E　DOS進入點與DISKⅡ界面電路圖 185

DOS 進入點 186
DISK Ⅱ界面電路圖 188
DISK Ⅱ類比線路板電路圖 189

附錄F　DOS命令總整理 191

符號意義 193
檔案名稱 195
管家命令 196
存取命令 201
循序主文檔的命令 204
隨機存取主文檔的命令 208
機器語言檔的命令 210

附錄G DOS 使用程序 213

BOOT DOS 214
INIT 磁碟 214
自意外的RESET返回原狀態 214
在程式中利用 DOS 214
建一TURNKEY 系統 215
建立及取出循序主文檔 216
將資料加入循序主文檔 217
經由循序主文檔控制APPLE 217
建立及取出隨機存取主文檔 218
複製主文檔 219
APPLESOFT的CHAIN功能 219

附錄H 更新 DOS 成爲16段落式 221

更換PROM 222
在語言系統上使用 DOS 223
建立一TURNKEY 系統 223

附錄I 使用 BASICS 磁碟 229

以BASICS BOOT 磁碟 229

附錄J 使用 FID 程式 231

進入 FID 程式 232
給定檔案名稱 232

FID 命令群 233
“錯誤”後之處置 236

附錄K　使用MUFFIN 程式 237

使用MUFFIN 程式 238
利用通用字元“＝” 240

第　一　章

組立與操作

機件分解
電纜線連接
控制板組立
多磁碟機系統的組立
二號磁碟機(DISK　II)及磁碟的保養
插入及拉出磁碟

機件分解

你的二號磁碟機系統由下列諸項所組成，你將發現有：

1. 磁碟推動器（最大的盒子即是），以下簡稱磁碟機。
2. 一個印刷電路板（控制板）可插到Apple Ⅱ的主機板架上。
3. 一帶狀電纜，此電纜與磁碟機聯接在一起。其功用爲聯接磁碟機與控制板。
4. 一系統主磁碟。
5. 一BASIC程式磁碟。
6. 此手册。

如果你只買了磁碟機（譬如你的控制板之第二架磁碟機）那麼你的系統將不會包括所有上列的項目。

如果你需寄送你的磁碟系統時（譬如你退還給經銷商或要送到工廠修理的話），應該保留所有的組件以便檢修。

＊＊＊特別注意＊＊＊

在聯接或斷接任何二號磁碟機或

Apple Ⅱ組件之前把電源關掉

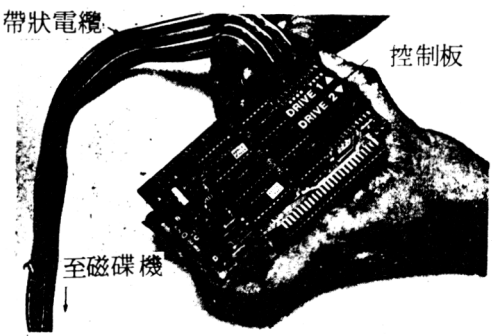
這是絕對必要的

電纜線連接

在使用時，我們要把磁碟機以一帶狀電纜連接到控制板上。帶狀電纜的一端已被固定在磁碟機上，如果這是你的第一個磁碟機的話，帶狀電纜就該被連接到控制板上方DRIVE 1 組接脚上。

＊　＊　＊　注　　意　＊　＊　＊

如果磁碟機和控制板之間的電纜插錯的話，極可能對磁碟機組件造成相當的傷害。又爲了要保證正確的組合，在控制板與Apple Ⅱ主機連接之前，必須要確定電纜已插入了控制板之內。另外還有兩個組立的重點。第一，不要把控制板和磁碟機之間的電纜擠得太緊。當電纜端正確無誤的插入控制板後，電纜就應儘量往控制板的一端疏展，如下圖所示。



把電纜連接到控制板上

第二，確定所有控制板連接器上的接脚已正確地插入帶狀電纜連接器相關的洞內。在把控制板裝入系統前做好這些連接工作，你應可以很明白的看出接脚接的對不對。

如果你是在把第二台磁碟機連到系統上去，你就該把帶狀電纜接頭連到控制板下方第二組接脚（DRIVE 2）上去。接的時候注意事項與第一台相同。

如果你想更新一個舊的磁碟作業系統（DOS）到最新的16段落型磁碟系統，你將必須更換控制板上兩個PROM，詳細資料請看附錄H。

控制板的組立

當用帶狀電纜把磁碟機與控制板連接完後，只要遵守下面步驟就可把控制板插入APPLE Ⅱ的插座上而完成磁碟機的組立：

1. 關掉在APPLE Ⅱ背面的電源開關，爲了防止計算機受損，這步驟是很重要的。如果在電源開的時候把控制板插入或取出，則極可能對控制板本身及對APPLE Ⅱ造成永久的傷害。
2. 把APPLE Ⅱ的頂蓋拿掉。只要把機器後端稍在上提就可辦到（機器後端是指與鍵盤距得最遠的那一端），此時不必再繼續往上提而只要往後推一下直到頂蓋完全脫離爲止。
3. 在APPLE Ⅱ主機內，電路板的後端，有8個瘦而長的插座叫做“溝槽”（slot）。最左邊的那個（從鍵盤這端看入）是溝槽零，最右邊的是溝槽7。把控制板放到溝槽6，也就是右邊數來第2個。其實除了最左邊的溝槽0外，控制板放到那個溝槽都沒關係。但是Apple 磁碟控制板的標準位置是在溝槽6，許多Apple 軟體（包括本手册）都是基於這個標準而寫的。
4. 在從計算機插入或拉出任何電路板時必須確定電源已關。然後把控制板上那像手指的金屬部份插入溝槽6中。插入後由於摩擦力的關係控制板就會安置得很穩。因爲這些手指狀金屬是用來做接點的，所以你最好不要拿手去碰它。在組立之前，你可用酒精去擦乾淨這些手指（也可擦擦你自己的手指如果你想要的話）。

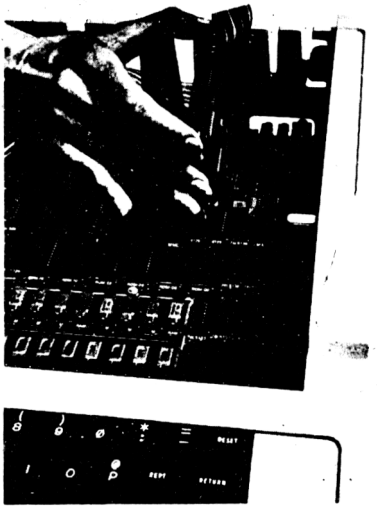


圖1－2　控制板插入

5. 調整帶狀電纜，使它平放及通過APPLE Ⅱ機盒後開口之間平坦部份，如下圖所示。當蓋子合上去時，就會把電纜夾住，如此就不怕盒外電纜受到拉力了。

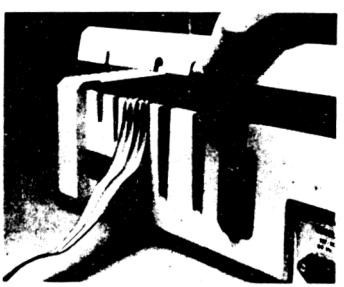


圖1－3　電纜放置法

6. 把APPLE Ⅱ的蓋子重新放回。記著先平移使得蓋子前端接合，然後壓下後端的兩個角，使它們接合好。
7. 現在二號磁碟機已接好，此時可把APPLE Ⅱ主機的電源打開了，把磁碟機放到一個便利的地方，通常是放在APPLE Ⅱ的旁邊或上面。

多磁碟機系統的組立

每一個控制板能夠接兩套磁碟機，其中一台連接到控制板上方的DRIVE 1 組接脚，另一台則連接到控制板下方的DRIVE 2組接脚。其中第一及第二台磁碟機可分別連到溝槽6中控制板的DRIVE 1 及DRIVE 2 ，第3及第4 台磁碟機連到溝槽5的控制板，第5及第6台磁碟機連到溝槽4的控制板，餘類推。

如果你使用多磁碟機系統，最好是把每個磁碟機的前端標上它的溝槽號碼及它是本系統第幾號磁碟機，因爲以後你寫程式時也需參考到這些編號。

二號磁碟機及磁碟的保養

二號磁碟機不像APPLE Ⅱ的主機一般，它是一種機械裝置，裏面有馬達及可移動的部份。因此它是一種較計算機更易損壞的機器。不小心的操作，譬如把磁碟機掉到地上或者有什麼東西擊到它都會使它操作不正常而故障。還有磁碟機最好也不要放在電視旁，因爲電視所發出之強磁場會對磁碟機產生不良影響。如果必須放在電視機旁時，也要至少保持2呎以上的距離才行。

軟性磁碟是一片小的（直徑約5吋）塑膠碟子，它的表面上有

磁性材料使得資料可以存入或消除。外面的這層磁性材料與錄音帶上的磁性材料相同。磁碟永久的被封在一個方的黑色塑膠套子裏，這套子可以保護磁碟，使它乾淨及讓它能夠自由旋轉。這種包裝永遠不需打開。

FLOPPY DISKETTE 這名稱的由來就是因這種磁碟有彈性。舊式計算機資料儲存元件所用的磁碟原理與這種磁碟相同，只不過它是硬的。雖然這種軟磁碟具有一些彈性，但如過度彎曲仍會對它造成傷害。磁碟上的覆蓋物含有潤滑劑及清潔劑以保護磁碟免於故障一所以對這層覆蓋物我們也要尊敬它。

永不要讓任何東西接觸到磁碟表面棕色或灰色的區域。操作時只能去動黑色塑膠套子部份。當磁碟不用的時候，把它保存在紙套子裏。這些紙套子能防止由於靜電場建立而去吸引灰塵。存放磁碟最好是使它們能立起來，爲此目的而設計的烯筆記簿型架子是很方便使用的。

磁碟能儲存非常大量的資料：一個磁碟能儲存1,146,000 比次(bits)的資料。單一的一個比次資料在磁檔上佔的位置實在很小。因此一個抓傷或一個手指印都能使資料發生錯誤。不要把磁碟放在髒的或油膩的地方，不要讓它們表面積上灰塵。

在寫磁碟的編號時，不能用太尖的筆也不能重壓。最好的辦法是先把編號寫在標織上，然後再貼到磁碟上去。

不要使磁碟接近磁場，也就是說不要讓它接近馬達或磁鐵等物，它們也不該放在電子裝置（像電視）之上。它們只能暫時的放在APPLE Ⅱ或磁碟機旁。

磁碟對溫度過熱或過冷都很敏感。所以不要把它放在太陽下或其它熱源旁否則就會使磁碟彎曲而使資料受損。在大熱天裏，汽車的儀器板或車身都會使磁碟因熱而損毀。磁碟到攝氏 51.7 度時還

能工作正常。磁碟受熱而損害第一個現象就是彎曲或是黑色塑膠部份隆起。

若有適當的保養磁碟的平均使用時間約爲 40 小時－這也不少了，想想看只需幾秒鐘就可把許多程式儲放到它裏面去。但是若有一點點的不小心，磁碟也將以不服侍你做爲報復。

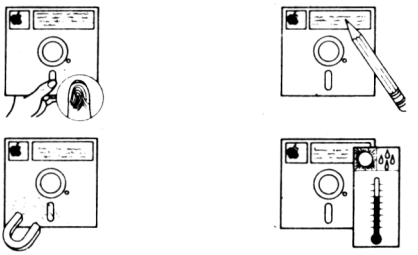


圖1 — 4

插入及拉出磁碟

使用磁碟機遠比使用卡式錄音帶要來得迅速與簡單，只是我們需多花一點功夫去保護磁碟。操作磁碟機時也需小心。磁碟機的門是要從下邊打開，然後再把磁碟平放滑入，有標籤那面朝上，如圖所示。有橢圓形那端先進入磁碟機，有標籤的那端在後。

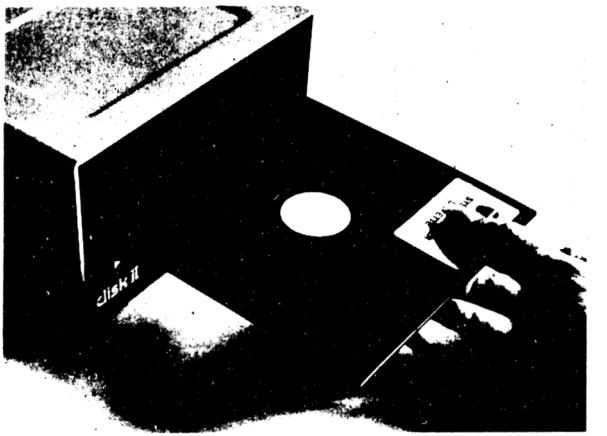


圖1－5　插入一片磁碟

＊　＊一項良好的法則＊　＊

握磁碟時，用你的右手姆指放在標籤之下，然後推入磁碟機，如此保證不會出錯。

慢慢的把磁碟推入磁碟機中，不要使磁碟扭曲。如果推得太用力，磁碟就可能會永久受損。把磁碟機門拉下關上，關上時，你可看到兩條手指狀金屬（在溝槽內），它們是當磁碟機門關上時清潔磁碟用的。

在移出磁碟時，首先打開磁碟機的門，小心的拉出磁碟。在打開磁碟機的門的同時，磁碟機也把讀寫頭舉起而離開磁碟。如果你想把一片尚未使用的磁碟放在磁碟機數小時之久，那麼最好把磁碟機的門打開，如此可避免讀寫頭長期的與磁碟接觸。

當磁碟機使用中的警告燈正亮著的時候絕對不可在此時移出磁

碟。否則將會永久的使磁碟受損，磁碟上的資料也必然受損。萬一發生這種事，即使磁碟仍可再使用，原來的資料也永遠恢復不了。

第　二　章

操作開始

背景
特殊鍵
把DOS程式從磁碟搬入主記憶體
如果BOOT動作失靈的話
新磁碟的啓始
DOS的載入及保存
登錄(CATALOG)
在一檔名稱內包含了什麼
重新命名檔案的方法
刪除檔案的方法
不小心重置機器時的修復法

背景

在學習使用磁碟及其內作業系統之時也包括了學習一些特殊的指令，裏面有幾個是直接從熟知的BASIC語言中推展而來。本手册假定你已清楚了APPLE Ⅱ及會寫一些基本的BASIC程式。

如想學如何使用APPLE Ⅱ及整數型BASIC語言，請參考APPLE Ⅱ BASIC程式手册（APPLE出版，編號 # A2L0005X）。如想學如何使用APPLESOFT型BASIC程式，請參考APPLESOFT Ⅱ程式手册（APPLE出版，編號 # A2L0006）。APPLESOFT手册內假定你已經熟知APPLE Ⅱ及簡單的BASIC語言。如果你兩本手册都不熟，你得先去學學APPLE Ⅱ再回來學本書所要講的 DOS 。

* * *

在這本手册內我們舉了很多程式的例子來說明如何使用 DOS，大多數程式是屬於APPLESOFT型，少數則是整數BASIC語言型，有時我們會說明爲何需要把一APPLESOFT型程式變成整數BASIC型程式，有時則不加以說明。詳細資料請看APPLE-SOFT 手册附錄M，那裏比較了整數BASIC型程式和APPLE-SOFT 型程式異同。

爲了得到一些經驗的傳授即使需讀很多書也是值得的。一旦當你把磁碟機裝好及把APPLE Ⅱ主機電源打開後，就要切實的遵守所說的步驟去試驗。

先把APPLE Ⅱ的語言置爲BASIC 。 可以是整數BASIC型或APPLESOFT型。把系統主磁碟放到磁碟機內。磁碟上的標號必須是0004-0002XX。最後兩個數字用X來代表，而它們是什

麼不怎麼重要。如果你有很多台磁碟機裝在一起，則用第一台。此節裏我們只處理一台磁碟機的情況，且假設你是依照慣例把控制板插到APPLE Ⅱ主機板第6個溝槽上的。

磁碟機連好了，磁碟也插入了磁碟機，磁碟機的門也關上後，你會發現APPLE Ⅱ執行程式時和沒有裝磁碟機前一模一樣。什麼也沒變，好像磁碟機不在那裏似的。就APPLE Ⅱ來說，磁碟機的確還沒有連接完畢：只差一個特殊指令來通知計算機有一架磁碟機及一堆 DOS 的新命令可以使用了。

雖然 DOS 命令和BASIC的命令看起來很像，但是不一定遵守同樣的規則。例如，幾個 DOS 的命令不能以逗點分開同時的排在一行上。如此做，計算機會通知你語法錯誤（ Syntax error）。

特殊鍵

本手册有時會以大括號把特殊鍵包起來以資識別。這些特殊鍵在APPLE Ⅱ的鍵盤上都找得到。

｛ RETURN ｝：表示你需按下“ RETURN ”這個鍵。在每個指令後需按下RETURN 鍵。

｛ RESET ｝：表示需按下“ RESET ”鍵。如果你有一個自動啓動的 ROM ，按下RESET鍵後計算機將會“嗶”一聲然後顯示一個催促信號（ Prompt Character）。如果是老式的 ROM ，按下“ RESET ”鍵後程式將會回到監督（ MONITOR）程式，監督程式以＊做爲它的催促信號。

｛ ESC ｝：表示需按下“ ESC ”鍵。“ ESC ”原來是脫離（ escape ）的意思，但現在它有另外的使用法。

{ CTRL }：和上面的鍵稍有不同。它表示你需按下“CTRL”（即control控制）鍵而且一直按著當你在按另外一個鍵時。譬如，{ CTRL } C表示你在按下C鍵的同時也按著CTRL 鍵。有時我們使用不同的符號：如 CTRL－C與{ CTRL } C代表著同樣一件事。

＊ ＊ ＊ 注 意 ＊ ＊ ＊

與 CTRL 鍵同時按下的鍵不會顯示在螢光幕上。

把DOS程式從磁碟搬入主記憶體（BOOTING DOS）

把 DOS 命令加到BASIC語言內去的步驟叫做“BOOT”磁碟。在整數型BASIC程式，APPLESOFT型程式，或在監督程式內都有指令可以來BOOT 磁碟。因此有許多方法可以達成BOOT 的目的。在整數型BASIC及APPLESOFT型程式中可用PR # s及IN # s 命令（參考你的APPLESOFT程式手册）。在監督程式內，使用 CTRL 鍵的“控制命令”就可達到目的。不管把 DOS 載入主記憶體的方式如何，所得到的DOS 程式是完全一樣。

在下面的例子中，小寫字母 s 代表控制板插入APPLE Ⅱ主機板的溝槽編號。控制板標準的插法是插入溝槽6（見第一章），在下列所將說明之命令之後，必須壓下RETURN 鍵。

整數BASIC程式（催促信號爲＞）

你能用下列兩命令任何之一來 BOOT 磁碟

PR # s　　例：PR # 6

或IN # s　　例：IN # 6

APPLESOFT型程式（催促信號爲]）

命令和上相同。

監督程式（催促信號爲＊）

可用下列兩命令之一來 BOOT 磁碟。

Cs0 0G　　例：C60 0G

或 s｛CTRL｝P　　例：6｛CTRL｝P

在本手册的其它部份，當遇到以上述方法來重新啓始 DOS 時，我們就稱這步驟爲“BOOT DOS”或“BOOT 磁碟”。以上兩種名詞代表著同一事物，在計算機使用者之間非常通用。BOOT 是靴帶（bootstrap）的簡寫。在任何情況下，它不是代表去踢磁碟，雖然你有時候心情眞的是想去踢它一脚。

現在我們來看看如何從系統主磁碟中BOOT DOS。首先，讓APPLE Ⅱ在BASIC語言型中一整數BASIC型或APPLESOFT型均可。再檢查磁碟有沒有放好。下一步鍵入 PR # 6 後按 RETURN 鍵（從此以後，我們假定在每個指令之後你都會自動按 RETURN 鍵）。

按下RETURN 鍵後，代表“使用中”的紅色燈泡會亮，這時磁碟機會很吵，但不要担心，它不會飛走的。在 10 秒鐘之內一個訊息會出現，這訊息大約是下列的形式：

DOS VERSION 3.3　　04／15／80

APPLE Ⅱ STANDARD SYSTEM MASTER

如果現在你想使用BASIC語言，你將會發現大部份的命令仍然管用。除了突然跑出上面的訊息外，使用APPLE Ⅱ似乎沒有什麼異狀。但實際發生了下列事：你現在有了一些新的命令，還有一些舊的命令現也有了新的能力。有兩件事情已經發生但你並沒有看到，即：

1. 指在你可使用的最高記憶體位置的HIMEN指標已被重置以來存放 DOS 程式。

2. 你的Apple Ⅱ計算機可能會失去一些產生高鑑別率圖形的能力。這當然要看計算機記憶體容量的大小。

** 13 段落型的 DOS 版本不能被 16 段落型磁碟系統所取出載入主記憶體。(磁碟只會在那裏旋轉及打嗝，但什麼東西也出不來)。此時我們可先用MUFFIN 程式把一 13 段落型磁碟轉換成16段落型磁碟。也可藉著 BASICS 磁碟之助把一 13 段落型磁碟資料載入你的 16 段落型系統中。見附錄 I 便知分曉。

如果BOOT動作失靈的話

如果你不能成功的 BOOT 出你的系統主磁碟的 DOS，再重讀讀你的手册，它可以把所有問題的 90% 醫治好。

雖然不太可能發生，但是如果你的系統拿雪曼型坦克或其它類似的東西運送，磁碟機內部的接點也許將會鬆動。而如果你又被磁碟機內部結構嚇到的話，代理商將會很樂意的幫你檢查。

如果你喜歡親自動手的話。首先，可把計算機電源關掉，然後把磁碟機與控制板分開，把磁碟機底部的 4 個螺絲轉鬆，然後整個內部機構可從機盒內提出。輕推連接器，使連接器與電路板緊緊接牢，再組合回去。現在可能已修好了，如果這急救失靈了，趕快去找你的代理商，而不要自已做任何的判斷去修理。

新磁碟的啓始 (INITIALIZE)

這本手册所指的系統主磁碟是一種非常特殊的磁碟：它有許多程式可以讓你複製整張磁碟，更新 (update)含有早先版本的磁碟，及其它。還有很多程式足以展示 DOS 的許多能力也包括在此磁

碟中，本手册將會一一討論它們。

把系統主磁碟從磁碟機中拿出，再拿一片空白磁碟插入磁碟機。現在來做一個實驗。使計算機能作BASIC程式後，打入：

PR # 6

看看會發生什麼事。首先，紅色“使用中”燈會亮，然後磁碟機會發出幾聲碰撞聲，接著磁碟機就安靜的輕輕的旋轉，且永不停止。現在你必須按下RESET鍵才能使它停止（正常情況來說，這不是個好主意。但是現在的情況並不正常）。在還沒有按下RESET鍵前最好把磁碟機的門打開，如此就可使讀寫頭先離開磁碟的表面。

剛剛所發生的事情是這樣的：你的APPLE Ⅱ在一片空白磁碟上開始了一次無結果且無終止的搜索一資料（如果此磁碟夠乾淨的話）。當一片新磁碟生產出來後，在它上面是不包含任何資料的。爲了使它在計算機系統內能夠運作，必須先放一些特殊的資料到磁碟上去：這就叫做磁碟的啓始設定（INITIALIZE）。

如果你是照著上面給的例子操作，你的空白磁碟現在應該在磁碟機內且你剛剛按下了RESET鍵。現在把此片空白磁碟拿出，再把系統主磁碟放入，把磁碟機的門關上。使計算機能作BASIC程式後鍵入：

PR # 6

你將得到和你BOOT磁碟時一樣的訊息顯示出來。而BASIC程式語言的命令中也被加入了 DOS 的命令。

INIT 命令能夠去啓始一片“奴”（Slave）磁碟。奴磁碟和主記憶體的大小有關：啓始一片磁碟的系統它的記憶體容量與能使用此片磁碟的系統記憶體容量大小有關。如果一片奴磁碟是由－32K 系統所啓始，那麼此片磁碟只能被32K（或更大）的系統所使用。如果在更大的系統內使用這片磁碟，只有32K 的記憶體

會被用到。在啓始一片奴磁碟後，你可用MASTER CREATE（主磁碟新創）程式（見第五章）來把奴磁碟改成主磁碟形式，因爲主磁碟的 DOS 是可自動重定位，如此記憶體就可以有效的運用。而MUFFIN 程式可以把 13 段落型磁碟資料轉換成 16 段落型磁碟資料。

INIT 命令需要利用到一個以BASIC語言所寫程式叫“招呼”（greeting）程式因爲它和你打招呼：每當你在 BOOT磁碟時它都自動的被執行。這招呼程式通常命令爲“HELLO”但是如果你高興的話你也可隨便給它一個名字。而在啓始磁碟時還是用標準名字爲佳，因這樣較簡單。

以下就是在啓始奴磁碟時一步一步來的詳細步驟，假定此時 DOS 已經被 BOOT 到記憶體內了。

1. 把主磁碟從磁碟機拿出，再放入一空白磁碟。
2. 鍵入“NEW”，然後鍵入一招呼程式。以下是招呼程式的一個簡單例子。

```
5  REM  GREETING-1 PROGRAM
10  PRINT "SLAVE DISKETTE CREATED
      ON 48K SYSTEM"
20  PRINT "BY AMY DOAKS ON 8 AUGU
     ST 1982"
30  END
```

圖2－2

你必須提供你自己的名字，系統大小，日期及其它情報以便快而簡單的查出磁碟的歷史及主／奴的狀態。你可也執行招呼程式來看看它是不是照著你所期待的結果出現。

3. 如果程式令人滿意，再鍵入這指令：INIT HELLO。當你壓下RETURN 鍵後，磁碟大約會旋轉1分鐘，偶而

會發出一點聲響。當 INIT 指令做完後，就會有催促信號出現在銀幕上（譬如：APPLESOFT的催促信號是]）。

4. 當磁碟機靜下來，"使用中"的燈也熄了之後，把磁碟拿出然後做上標記。標記應有如下的格式：某年某月某日所建，容量X之主（或奴）磁碟。這樣你只要看標記就知道它不是空白的。

把APPLE公司所提供的系統主磁碟好好儲存起來，放在一不會被熱或壓損傷或磁碟影響的地方。也不要弄掉了。你必須非常小心的保存它，因爲它包含了很多有用的程式。

一片磁碟被啓始後，它就成爲一片可使用的奴磁碟。當要標記它時，需把它從磁碟機拿出。標記好後再把它放回磁碟機中，試著把它 BOOT 出來。鍵入 PRINT後螢光幕上可得到如下的訊息（如果你是照上面步驟2所鍵入的）：

```
SLAVE DISKETTE CREATED ON 48K SYSTEM
BY AMY DOAKS ON 8 AUGUST 1982
```

因爲這一度是空的磁碟現已能做 BOOT 的動作，可知它已經被啓始成功了。從現在開始，你可用這新啓始之磁碟來作各種不同的實驗了。而在系統主磁碟上，由於它是"禁寫入"的，下面有些步驟就不適用，詳見第四章。

如果你購買了一批磁碟的話，先啓始幾片備用將是個很好的主意。

用DOS系統來載入及保存（LOAD及SAVE）

用啓始好的磁碟把系統 BOOT 入。先鍵入：NEW，以確定主記憶體沒有程式在內。這將從記憶體中洗掉你的招呼程式（招呼程式就是在BOOT DOS程式時被載入及執行的程式），而磁碟中

仍保有它。

現在鍵入下面這簡單的程式：

```
5  REM   COUNT PROGRAM
10  FOR I = 1 TO 10
20  PRINT I
30  NEXT I
40  END
```

執行它一兩次以確定合於所求。在APPLESOFT中執行完這程式時你將看到：

```
1              2              3
4              5              6
7              8              9
10
```

爲了參考方便，可稱這程式叫“ ONE TO TEN ”程式因爲它是從 1 數到 1Ø 。要把程式儲放到磁碟上，鍵入下面指令

SAVE ONE TO TEN

按下RETURN 鍵完成這指令後，磁碟將會旋轉一陣來使此程式保存在磁碟內。

如果你只鍵入 SAVE 而不給任何的名字，此程式將被保存在錄音帶上（假定你已照著 BASIC程式手册內所說裝好錄音機的話）

爲了確定程式已被保存在磁碟上，做以下的步驟：

首先，鍵入： LIST ，然後 RUN 。來看看此程式是否仍存在主記憶體中及能否正常操作。結果是，用 DOS 系統把程式保存到磁碟後將不會影響主記憶體中任何程式的操作或執行。

現在，鍵入 NEW ，然後 LIST 。結果此程式在主記憶體中不見了－此乃 NEW 命令的關係。

好，再鍵入LOAD ONE TO TEN ，磁碟機大約會響兩分鐘，然後鍵入到印的指令，程式又出現了。執行它後你可發現它完

全正確。這就是使用磁碟保存或載入程式的方法。它和使用錄音帶的方法幾乎相同（除了必須給程式的名字外，但也因爲如此，它有較快的速度）。

登錄(CATALOG)

你已把“DNE TO TEN 程式放到磁碟上了，事實上，你也一起存放了另一個程式。爲了看看什麼程式存放在磁碟內，鍵入 CATALOG 命令，此令一下，所有在磁碟上程式的名稱都會列印出來，如果你的程式以整數BASIC寫的話，現在你將看到

```
I  002   HELLO
I  002   ONE TO TEN
```

在左邊的I字母表示此程式是以整數BASIC所寫，如果是以APPLESOFT所寫，你將看到“A”字母。除了BASIC程式檔外，還有許多其它種類的檔也可被存入，這些將在第6章至第9章解釋。在檔型後面的數字代表所儲程式的長度，在此例中，002個磁碟段落被用來儲存此程式。最短的檔，即空檔（見第6章），需要一個段落來儲存“管家”所需的情報。總括來講，一片磁碟總共可以儲存496段落。最後，在目錄之每一進入點包含著程式的名稱。有關情報如何被儲存在磁碟上的詳細資料見附錄C。

當一個檔長度超過255個段落時，CATALOG 命令報告這檔的長度時將從再從000開始算。

當你在看程式目錄時，你無法分辨何者是自己的程式何者是招呼程式。所以把每個招呼程式的名字都取得一樣是一個很好的主意

有時你磁碟上程式數目太多，一個螢光幕內容不下。這時 CATALOG 命令將只使前 18 個程式名稱列出，當你準備要看磁碟上其它程式名稱時，你可去按除了RESET，CTRL及SHIFF以外的任何鍵就可達成目的。

在一檔名稱內包含了什麼

檔名稱長度可以從1到30個字元長（character），如果超長 DOS 將會把它切除成 30 字元長。檔名稱第一個字元必須是字母，任何可印出的字元（除了" "）都可出現於檔名稱之中。

下面是一些合法的名稱：

SOMNAM BULISTICS

ONE TO TEN

HIRES 34

THE QUALITY OF MERCY: UNSTRAINED

而以下則是一些不合規定的名稱 （理 由）

1 TO 10 （數字開頭）

HI THERE, BABY （有" "符號）

INEPT EXCESS VERBIAGE DISQUALIFIES

NAMES （長度過長）

當登錄命令執行後雖然上面最後一檔的名字已被截成 30 字，但是當你在載入及執行程式之時，你仍可使用程式原先的長名字，一切仍將正確運行。

在目錄內的任一行都表示一個檔，你所存的BASIC程式就是一個檔。對檔名稱所做的規定同樣適用於程式的名稱。

如果一控制字元意外地（或故意地）打入名字之中，在執行登錄命令時那個控制字元將不會出現在銀幕上。譬如，當你想鍵入AGATHA 這個名稱之時，在鍵入T時不小心鍵入了{ CTRL } T，那麼在目錄列表時將成AGAHA。但是當你想從磁碟載入這程式時，你鍵入LOAD AGAHA，計算機將回答FILE NOT FOUND，即使你打入的名稱“似乎”和目錄內的名稱一樣。因此要小心：千萬不要不經心地把控制字元打入檔名稱內。（雖然，嘿嘿，如果有個秘密控制字元藏在檔名稱裏你就不能來取用我的戶頭了……）。附錄F中有關檔名稱的那一段包含有許多技巧，告訴你如何找出隱含在檔名稱內的控制字元。

重新命名檔案的方法

偶而，你也會想要把你的檔改一個名字。假定你覺得ONE TO TEN 這個名字太長而想改成COUNT，只要打入RENAME ONE TO TEN, COUNT就可。等磁碟機騷動一陣之後你將看到BASIC程式的催促信號。再打入CATALOG 來檢查是否照所預期的執行。

RENAME 命令並不去檢查你所用的檔之新名字是否原先已經存在。所以有可能經過幾次RENAME 後所有在磁碟上的檔都有相同的檔名字……。你最好避免此種狀況。

刪除檔案的方法

要把某一檔從磁碟除掉也很簡單，先打入：CATALOG 你再

次可看到磁碟上有兩個檔，再打入：LOAD COUNT（假定你已替ONE TO TEN換了名字），要刪掉COUNT的檔只需要打入：DELETE COUNT。再打入CATALOG 看看是否把COUNT檔給刪除掉了。

現在只有招呼程式（名字也許是叫HELLO）留下來了。因爲COUNT程式還在APPLE主機記憶體內（記得不？剛剛你LOAD COUNT了），你可再把它放回磁碟，使用：SAVE COUNT 命令。再看看目錄，你將發現COUNT檔已回去了。

如果你想刪掉一不在磁碟上的檔，系統會告訴你：FILE NOT FOUND的訊息。

不小心重置機器時的修復法

假定你是在整數BASIC或APPLESOFT型程式中且未使用DOS，如果你不小心按了重置鍵，你可用CTRL—C 來恢復你的程式。DOS 也有些恢復步驟，它們經常是被用來保存你的程式及資料。

如果你使用的是老式 ROM 監督程式且已BOOT DOS，在不小心按了RESET鍵後，你將收到（*）監督程式催促信號，爲了要回到 DOS 及BASIC程式內，打入3DØG。注意D和G之間是數字零而非文字O。

如果恢復到 DOS 的狀態沒有成功，但程式仍可列印，這時所有資料都還未失去可用下法：把程式先存到磁帶上去(要磁帶推動機的理由之一就是要拯救此一緊急情況）。然後以後你可BOOT DOS，把程式從磁帶載入記憶體，然後再保存到磁碟上去。

STOP 如果當磁碟機使用中的紅燈亮著時，你不小心（或故意的）按下RESET鍵，這時磁碟上的資料可能會被損壞，特別是在你使用SAVE，BSAVE或WRITE 等命令時最有可能。如果磁碟因此而損害，你的程式可能就無法恢復。所有方法都不成功時，你可重新啓始（re-initialize)此片磁碟後再重新使用，但是重新啓始後所有在磁碟上的檔都已被破壞。

如果使用中的紅燈繼續亮了數分鐘之久而此時你也沒聽見磁碟機的轉動聲，這時系統可能已被“懸浮”（SUSPEND）起來。這時只好再按下RESET鍵然後才可重新使用此系統。

磁碟可能會部份損壞而已，但也因此不能 BOOT 資料進主機中。然而，在此情況下，你有時能夠先以另一磁碟BOOT DOS，然後從那部份損壞的磁碟內把程式（或檔）載入主機再保存到另一未損害的磁碟，或者使用 FID 程式把個別檔或程式複製到另一片磁碟上去。

第　三　章

練習使用選擇權

磁碟機，溝槽及磁碟册的選擇

語法

啓始命令（ INIT ）

載入、執行及保存命令

（LOAD' RUN及SAVE ）

删除命令（ DELETE ）

命令綜觀：BOOT.CATALOG,SAVE,

RUN及DELETE

語言間的轉換：FP及INT命令

在程式中如何使用DOS命令

磁碟機，溝槽及磁碟册的選擇

大多數 DOS 命令讓你有很多種的選擇，例如你可選擇要用那個磁碟機，此磁碟機控制板要放到那個溝槽，及使用編號爲多少的磁碟册等等。

磁碟機的選擇即表示你可同時操作一架以上的磁碟機，每一片控制板可控制一至二架的磁碟機。正常時命令都是對第一架磁碟機（DRIVE 1）而發出，這是自然值（default）：即如果你不指定那一架磁碟機時，系統就當做是第一架磁碟機。如果你想要用第2架磁碟機，你就需在檔名字後面（或其它選擇後面）以逗號分開鍵入D2 。例如，要啓始第2架磁碟機之磁碟，你應用如下的命令:

INIT HELLO, D2

在指定第2號磁碟機以後，再打進來的指令就都是針對第2架磁碟機的（直到再重新指定到第1架磁碟機），因此第2架磁碟機現在就成爲系統的自然值。鍵入上述啓始指令後，再打入：CATALOG ，此命令就會把第2架磁碟機所存之檔名字全部列印出來。若現在又要指定使用第1號磁碟機，同樣的打入：CATALOG，D1 （用D1 指示使用第1號磁碟機，並以逗點與命令分開。）這時第1號磁碟機上之磁碟所含檔案的名字都會列印出來，而系統之自然值也同時轉變成第1號磁碟機。

如果有兩架以上的磁碟機被使用，那麼就需要增加控制板，這些控制板就需放到不同的溝槽上去。你可以下法指定溝槽n（n從0到7）：用逗號分開把 Sn 寫到指令（或其它元件選擇符號）之後。例如，要啓始位於溝槽5的控制板上的第一架磁碟機，你可用：INIT HELLO, S5, D1 來指定。注意，檔名字要先寫，其它元件選擇符號的次序則可任意排列。

溝槽的系統自然值就是你在DOOT　DOS　時所用溝槽的編號，等到另外一個溝槽被指定後，那個新指定溝槽的編號就成了系統的自然值，直到指定溝槽編號再次更改時爲止。

如果使用一個附有選擇溝槽的 DOS 命令但是在溝槽內又找不到控制板時，那麼你將得到：I／O　ERROR 訊息，其它似乎未變，然而 DOS 現在已把溝槽自然值就設定在這無控制板的溝槽上，而命令就在這不存在的磁碟上“執行”起來。即使你下個命令指到正確的溝槽，它也須等到這不存在的磁碟執行完命令才能繼續，這樣就永遠的陷在那邊了。如果在主記憶體並沒有什麼程式需保留下來的，這時只要重新BOOT　DOS即可。但若是你想恢復你的程式，可作下面這些動作：

1. 重置溝槽自然值。打入：CATALOG，　Sn　S是正確的溝槽編號。
2. 當這系統懸浮不動時，按下RESET鍵。
3. 打入3DØG　一切都將恢復正常。

BOOT　DOS　一定是從第1架磁碟機而非第二架磁碟機來執行。

册編號選擇是用來防止磁碟不小心的被寫入。例如，假定你有一個存貨系統的程式存在磁碟內，每月份的記錄都存於一有特殊册編號不同的磁碟上面。那麼，當你想對某月輸入情報時，你必須指定出正確的册編號才可，否則你的輸入將不會寫入磁碟而且你將得到：VOLUME　MISMATCH 的訊息。

當磁碟啓始時就可以指定它一個册編號。只須在命令後（或其它元件選擇符號後）以逗號分開打入－Vn　。例如，啓始一片磁碟

，它的招呼程式名字叫“ START UP”，而這片磁碟位於第5個溝槽控制板的第2架磁碟機，並指定它的册編號爲128，你可打入下面命令：INIT START UP，D2，S5，V128

一磁碟的册編號除非重新啓始該磁碟否則將永不會被改變。

磁碟機編號，溝槽編號及册編號的選擇語法可以任意的次序來寫，如以上面例子的命令爲例和下面寫法是一樣的：

INIT START UP，V128，S5，D2

INIT START UP，S5，V128，D2等等。

一個磁碟的册編號必須在1到254之間。如果在磁碟啓始時沒有指定的話，那麼其自然值就是254。

以下之命令：INIT HELLO，V0 並沒提供任何一點訊息。而系統將給磁碟自然值册編號254。（0不屬於1～254之間。）

所有的DOS命令後你都可以加上册編號，(如果你希望DOS系統在執行命令時幫你檢查在磁碟的册編號與所加V選擇符號內所指定的是否相符的話)。假使你的命令不指定任何册編號，或指定0册（V0）或V後面沒有數字，DOS將不理會這些“册編號”。如你不小心指定了一個錯誤的册編號，系統將以：VOLUME MISMATCH 訊息來通知你。

當要求看目錄時（即當你打登錄命令時），不會有册編號不合的錯誤訊息產生。如果你想知道一片磁碟的册編號，它就在目錄列表的第一行上。

其它選擇符號在討論各種命令時將會一一介紹，並且每個命令的語法都簡潔的整理在後面命令摘要附錄上。下面幾節就是要說明

這些簡潔的命令之含意。

語法

語法就是計算機命令的結構、次序及命令各部份正確的格式。它是一個簡單的符號用來表示每一個 DOS 命令。其中在中括號（[ ]）中的項目表示可有可無；元件選擇符號可以任意次序書寫；大寫字元及逗號必須照著所寫格式打入；小寫字元則代表你須提供的項目，有下面數種：

f　檔名字

d　磁碟機編號－1 或2

s　溝槽編號－1 到7

v　冊編號－通常1 到245 。

一磁碟之冊編號不可以爲 0 ，在一磁碟命令上指定冊編號爲 0 就是告訴 DOS 忽略磁碟上的冊編號，是一條“野蠻”的命令。

本手冊其它的簡寫符號被摘錄於命令摘要附錄的開始部份。

DOS 命令中任何的數字常數（如磁碟機編號，冊編號等等）都能夠以 16 進位表示，這時只要在數字前面加個 $ 符號即可。

啓始 (INIT) 命令

INIT 命令的文法如下：

INIT f [, Vv] [, Ss] [, Dd]

中括號代表裏面項目可有可無。

例：INIT HELLO, V17 , D2 能夠被解釋如下：

此命令叫做 INIT(需大寫），必須照著以上字元格式書寫。小寫字母 f 代表檔的名稱，現被合於規定的名稱“HELLO”所取代，下一個，選擇册編號出現了，你須先打一個逗號，再打入一個大寫字母V，小寫字母 v 表示册編號（在我們例子裏是 17 ）。接著，由於 Ss 項被中括號包著，因此這項可有可無，在我們的例子中此項被省略了，而 DOS 將會拿自然值當作溝槽編號。接著是磁碟機編號，它也包括在本例中，如前一段，先打一個逗號再加個D，小寫字母 d 被 2 所取代，代表選擇第 2 架磁碟機。

載入、執行及保存（LOAD,RUN AND SAVE）命令

DOS 的載入，執行及保存命令與使用錄音帶有類似的操作方式（除了在磁碟上可直接用檔名稱叫出程式）。然而在速度上磁碟最少快 10 倍以上，同時也可省掉按開關，錄音或倒轉的動作。一切完全自動。磁碟還有其它好處：例如程式名稱製成目錄及程式執行不需使用者介入等等。保存資料（到正文檔上去詳見第 6 章）也非常的簡單。在附錄 J 的 FID 程式還提供你一些其它的方法來處理各種檔案。

繼續用卡帶（錄音帶）系統來作程式的交易及儲存重要的程式及資料當做備用還是個非常好的主意。（雖然據經驗顯示，在儲存程式及資料方面，磁碟比卡帶有更高的可靠度）。

如果你的程式是以BASIC寫的，名字叫作HENRY，那麼這命令：SAVE HENRY 將會把程式放到磁碟上去，如果你有一架以上的磁碟機，HENRY這程式正常的話會被儲放在你BOOT DOS的那片磁碟上。在各種 DOS 命令中你能夠指定磁碟機編號、溝槽編號及册編號就如在 INIT 命令中所做的。例如，要把一名叫

AGATHA 的檔存放於位於溝槽 2 控制板的第一架磁碟機上，其册編號爲 214 ，那麼你需用這命令：

SAVE AGATHA, D1, S2, V214 和以前一樣，三種選擇元件符號可以任意次序排列。如果你省略了册編號值，那麼，希望天保佑你不會放錯磁碟，因爲此時 DOS 不會替你檢查所放入之磁碟編號是否爲 214 。

程式的名稱就是檔的名稱，因此必須遵守檔名稱的規定：它必須最長 30 字元及開始第一個字元一定是字母，它不得含有逗號及控制字，以下是合規定的程式名稱：

CHECKBOOK

THE QUALITY OF MERCY

HIRES 34

NOW : HEAR THIS!

要載入記憶體一名叫AGATHA 的程式，可用下面命令：

LOAD AGATHA

如果目錄中有此程式的名稱，此程式將被載入記憶體中。爲了試試AGATHA 程式是否眞的進入記憶體了，可看看它是否在螢光幕上會跑出一行標頭來。如果你希望AGATHA 載入後能被執行，當然，你能試試下面步驟：LOAD AGATHA然後 RUN 。但是還有一種方法只要一步就夠了：RUN AGATHA 這也是一條 DOS 的命令，它首先把AGATHA 叫入記憶體中然後執行它。

以下是SAVE，LOAD及RUN 命令之語法：

SAVE f [, Ss] [, Dd] [, Vv]

LOAD f [, Ss] [, Dd] [, Vv]

RUN f [, Ss] [, Dd] [, Vv]

舉例如下：

```
SAVE OUR HAPPY HOME, D1, S7
LOAD UP
RUN AMOK, S7
```

當你想保存（ SAVE ）你的程式到磁碟時，若得到一語法錯誤的訊息，原因不外你的鍵入發生錯誤或者是沒有BOOT DOS。首先，重新打入你的命令，如果曾經已完成BOOT DOS 了，那麼使用3DØG 試著來恢復它。如果以前忘了BOOT DOS，那麼千萬不要現在去BOOT DOS，因爲一BOOT DOS，在記憶體內之程式通通都會被洗掉。首先把程式存到卡帶（錄音帶）上，用 SAVE 命令。然後再BOOT DOS，接著再使用剛才的卡帶，用 LOAD 命令把程式載回記憶體。現在你就可以把程式保存到磁碟上去了。

如果①磁碟遭受損害（譬如有人把它夾在筆記本內）或②沒有啓始；③磁碟機內沒有磁碟，④磁碟機門開著的話，當你用 DOS 想要載入記憶體或保存程式到磁碟時，系統會給你 I／O ERROR 的訊息。查查看以上所列的原因並做修正，你並不需要重新BOOT DOS ，再試著去載入（或保存）一次即可。

如使用這命令LOAD HENRY而HENRY 不是磁碟機上磁碟裏一個程式的名字，那時將會得到如下的訊息：

```
FILE NOT FOUND
```

查看一下磁碟內目錄看看程式的名稱到底是什麼。所有的字元與間隔必須被鍵入得正如在目錄上所顯示出的一般。

刪除（ DELETE ）命令

如果想刪除磁碟上已有而你不再需要的檔，可打入DELETE命令。其語法爲DELETE f [, Ss] [, Dd] [, Vv]

例如，DELETE EXCESS V34, D2, S1 此命令刪除了一個名叫EXCESS檔，册編號爲 34 ，位於溝槽1的第2架磁碟上。磁碟上的段落只有在做刪除命令時才會空出來。

命令綜觀：BOOT. CATALOG.SAVE.RUN及DELETE

假定你是在執行整數BASIC型程式且系統主磁碟也在的磁碟機中，以下就是在APPLE Ⅱ螢光幕上的對話，你所打入的那部份下面劃了一條線（在 TV螢光幕上並不如此，注意）。首先打入 ＞PR # 6 來清除螢光幕。你將看到下面的顯示：

```
DOS VERSION 3.3                 04/15/80
APPLE II STANDARD               SYSTEM MASTER
>CATALOG
DISK VOLUME 254
*A 006 HELLO
*I 018 ANIMALS
*T 003 APPLE PROMS
*I 006 APPLESOFT
*I 026 APPLEVISION
*I 017 BIORHYTHM
*B 010 BOOT13
*A 006 BRIAN'S THEME
*B 003 CHAIN
*I 009 COLOR DEMO
*A 009 COLOR DEMOSOFT
*I 009 COPY
*B 003 COPY.OBJ0
*A 009 COPYA
*A 010 EXEC DEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
*A 028 LITTLE BRICK OUT
*A 003 MAKE TEXT
*B 009 MASTER CREATE
*B 027 MUFFIN
*A 051 PHONE LIST
```

```
*A 010 RANDOM
*A 013 RENUMBER
*A 939 RENUMBER INSTRUCTIONS
*A 003 RETRIEVE TEXT

>NEW
>10 PRINT "JABBERWOCK"
>20 END
>SAVE DEMO
WRITE PROTECTED
```

← 在此處，你應插入事先已啓始好的奴磁碟，它的寫入必須未受到保護。

```
>CATALOG
DISK VOLUME 254
 I 002 HELLO
 I 002 COUNT
>SAVE DEMO

>CATALOG
DISK VOLUME 254
 I 002 HELLO
 I 002 COUNT
 I 002 DEMO
>NEW
>RUN
*** NO END ERR
>RUN DEMO
JABBERWOCK
>DELETE DEMO
>CATALOG
DISK VOLUME 254
```

語言間的轉換：FP及INT

如果你一直使用整數型BASIC寫程式，突然想改用APPLESOFT來做的話，或著想以浮點算術來計算數字時該怎麼處理？首先，叫出APPLESOFT而不去驚動 DOS 。這時只要打入FP（這樣就好）幾秒鐘後APPLESOFT就醒了並可開始執行程式。FP 代表Floating Point(浮點算術) 。如果因爲某種原因APPLESOFT不存在的話－即不燒在韌體內又不在所用的磁碟內，此時LANGUAGE NOT AVAILABLE 的訊息將會顯示出來。FP 命令的語法如下：

FP [, Ss] [, Dd]

磁碟機及溝槽的選擇性使你可以指令一含有APPLESOFT的磁碟。

如果你一直都是在使用APPLESOFT及 DOS 的話，這時只需打入 INT（代表Integer BASIC ，整數型BASIC），就能使 DOS 和整數型BASIC連在一起。INT 的語法只有 INT 三個字元而已。如果你打入D或S之類的選擇元件符號時，你就會得到：SYNTAX ERROR 的訊息。

如果打入 INT 而你也正使用整數型BASIC，這時你將失去你在記憶體中所有之程式。同樣的，如果在使用APPLESOFT時打入FP ，你也會失去在記憶體中所有的程式。當你從整數型BASIC轉換成APPLESOFT型時（反之亦然），在記憶體中存在的程式都將失去。

除了直接在BASIC程式間可以互相轉換以外，你也可以進入監督程式及使用 DOS 命令後來達到這目的（不管從整數型BASIC或APPLESOFT進入監督程式都可），這時打入CALL−151 ，然後在螢光幕上就會有監督程式的催促信號＊出現。如果你想回到原先的BASIC程式（即和 DOS 連接的那一個），只要打入 3DØG 就行。

從監督程式中，你也可打入 INT 來把程式轉變成整數型BASIC或者是FP 把程式轉換成APPLESOFT型。上述兩種情況中，除了 DOS 程式仍將存在外，其它在記憶體中的程式全將消失。

當你想執行FP 命令時，而得到了一個PROGRAM TOO

LARGE的訊息時，先打入 INT 然後重置（ RESET ）整個系統，再打入FP 。

假使你的磁碟內有一個整數型BASIC的程式名字就叫作APPLESOFT，這時千萬不要打入RUN APPLESOFT的命令。如果你這樣做的話，APPLESOFT程式似乎會執行得很好，可是，譬如在執行當中按下RESET鍵，並想再回到APPLESOFT程式上去，這時你就有麻煩了，因爲此時 DOS 認爲你是在整數BASIC型程式上。

爲了要把這個名叫APPLESOFT程式從一片磁碟移至另一片，方法仍和以前相似：只要打入LOAD APPLESOFT 不管此程式在那一片磁碟上。然後把你希望將APPLESOFT程式保存住的磁碟插入磁碟機中，打入：

SAVE APPLESOFT 即可

在程式中如何使用DOS命令

在－BASIC程式當中若能使用 DOS 的命令將是非常好的一件事。例如，你希望磁碟中的招呼程式裏能使用CATALOG 命令以便把在磁碟上所有檔名稱列印出來即是。事實上，在BASIC程式中能用到非常多的 DOS 命令。它的格式是PRINT後接著一列字串，字串內包含了CTRL-D 字母及 DOS 命令。

以下是一APPLESOFT型程式，它是一個招呼程式，執行後，20 行及 30 行所要印的會先顯示在螢光幕上，然後各個檔名字會接著列印它們下面。（由於 40 行的CATALOG 命令的關係。）

```
5  REM   GREETING PROGRAM
10 D$ =  CHR$ (4): REM   CHR$(4)
     IS CTRL-D
20  PRINT "SLAVE DISKETTE CREATED
      ON 32K SYSTEM"
30  PRINT "BY AMY DOAKS ON 8 AUGU
     ST 1980"
40  PRINT D$;"CATALOG"
50  END
```

在APPLESOFT型程式內使用 DOS 命令的方法展示於上面程式中。首先以 CHR $ 函數定義了字串變數D $包含CTRL—D字元（第 10 行），在第 40 行就用到了此字串變數，注意，在D$ 後所用的半分號(;)及 DOS 命令需以＂　＂符號括起來。其實D$ 後的半分號是可有可無的，當你想節省打字時間及記憶空間時，儘管可把它們省掉，因此第 40 行可成爲：

40 PRINT D$ ＂CATALOG＂

在APPLESOFT型程式裏，可用 CHR$ 函數來定義CTRL—D字元如　10 D$ = CHR$(4) ：REM CTRL—D

上面因爲ASCII碼中，4 代表CTRL—D 字元，故後面加個註解（REMARK ）來說明它。（在整數BASIC型程式中則沒有CHR$ 這種函數）。

在整數BASIC型或APPLESOFT型程式中，你都可定義CTRL—D ，方法如下：先打入D$ =＂4 個字元，再把 control 鍵及D鍵同時按下，最後再打入引號＂即可。注意，CTRL—D不會顯現在螢光幕上，因此螢光幕上最後的樣子是D$ =＂。因爲控制字元不印在螢光幕，最好在每一指令之後附上一註解（REM ）來說明在引號內之字串是何物。上一個程式以整數BASIC型程式寫出就成了下面的樣子：

```
10 D$="": REM  THERE IS AN INVISIBL
   E CTRL-D BETWEEN THE QUOTES
20 PRINT "SLAVE DISKETTE CREATED ON
    32K SYSTEM"
30 PRINT "BY AMY DOAKS ON 8 AUGUST
   1980"
40 PRINT D$;"CATALOG"
50 END
```

在－PRINT的敍述中只能使用－ DOS 命令。PRINT指令必須以CTRL－D 起頭，以 DOS 命令結尾。

使用"→"字元來複製BASIC程式，若此程式包含有螢光幕上看不到的控制字，複製的程式隱含的控制字將消失。

若在程式中來執行 DOS 命令，則D$ 之前必須有－RETURN字元否則D$ 將不被理會。執行以下程式後：

```
5  REM   TESTCATALOG PROGRAM
10 D$ = "": REM   THERE IS AN INV
     ISIBLE CTRL-D BETWEEN THE QU
     OTES
20  PRINT "TEST";
30  PRINT D$;"CATALOG"
40  END
```

將會使得TESTCATALOG 顯示於螢光幕。此乃因爲第 20 行後的半分號(;)把RETURN 取消掉了。爲了改正此點及使CATALOG 命令能被當命令執行，只要把第 20 行後的半分號去掉即可。

下面這些 DOS 命令只能用在程式PRINT 述語內時且應以CTRL－D 開頭：

OPEN

APPEND

READ

WRITE

POSITION

下面 DOS 命令能獨立使用（直接執行模式）或用在程式命令 PRINT 以CTRL—D 開頭的述語內：

CATALOG	BSAVE
SAVE	BLOAD
LOAD	BRUN
RUN	EXEC
DELETE	CLOSE
RENAME	CHAIN
LOCK 及 UNLOCK	PR#
MON 及 NOMON	IN#

DOS 命令中的MAXFILE 在APPLESOFT 型程式中可以如上所述方式使用，但在整數BASIC 型程式內卻有特殊的用法。此將於第7 章EXEC 命令那一節中詳細討論。

DOS 命令中的 INIT 命令只能獨立使用（即立即執行模式），如果你忽視了以上警告將有非常不良的後果。

第　四　章

安全的操作

建出一"轉動鎖"(TURNKEY)系統
鎖制與解鎖(LOCK與UNLOCK)
查證命令(VERIFY)
磁碟的禁寫入
防止你遭到不幸的方法
複製(COPY)程式使用法

已經敍述過兩種可使你自己及你的磁碟免於受損的方法：第3章告訴你使用册編號來保證所使用的磁碟正是你所要用的。第2章告訴你在檔名稱內使用控制字元也可當作保護程式的一種方法。（見第2章，“在一檔名稱內包含了什麼”這一節，也可參考附錄F，“檔名稱”那一節。譬如在目錄中有“MY BANK ACCOUNT”這一程式名稱，在這名稱中的某一處你可預先放置一個控制字元，那麼別人就不可能去取用這程式了。

本章要告訴你各種不同的方法去防止各種不同事件侵害你及你的磁碟，遲早你會發現這些技巧對你很有幫助。首先，考慮製作一個特殊用途的“轉動鎖”（TURNKEY）系統。

建出一轉動鎖系統(TURNKEY SYSTEM)

假如一位醫生想在APPLE Ⅱ上作些會計工作。理想情況下，工作人員只要輕鬆地打開開關，打入｛RESET｝6｛CTRL｝P｛RETURN｝後馬上可以進入這位醫生的會計程式中。因爲這會計程式能與使用者以英語通訊，因此工作人員並不需要知道BASIC程式語言或任何有關APPLE Ⅱ的事。整台計算機系統都成了一個會計系統，計算機的內部特性對工作人員來講並不重要，他們只需知道如何來使用這會計程式就行。

這就是一“轉動鎖”系統的必要特性：從使用者的眼光來看，整架計算機只是執行某一特定工作的儀器，而使這特定工作系統開始工作也很簡單，簡單得像在一個鎖鑰內轉動一把鎖一般。在上例中，這個鎖就是把開關打開，然後在鍵盤上再按5個鍵而已，這些動作都不是需要計算機專家才能做得來的。

你能夠使用磁碟中的“招呼程式”（這程式當你在啓始時就必

須定好名字）來把APPLE Ⅱ系統轉變成－“轉動鎖”系統。例如假定你希望每當BOOT DISK Ⅱ時計算機執行COLOR DEMO程式（由系統主磁碟內提供）。以下就是如何來作：

1. 啓始－空白磁碟，方法如第二章所述。
2. 把系統主磁碟放入磁碟機內，打入RUN COLOR DEMO 如果程式執行無誤令你滿意後，打入｛ CTRL ｝C來把程式停掉並回到BASIC程式型中。
3. 把你新啓始好的磁碟放入磁碟機內。假定在啓始磁碟時你把“招呼程式”命名作HELLO。
4. COLOR DEMO 程式現在主記憶體中，如你打入SAVE HELLO， DOS 將會把原先叫作HELLO的招呼程式洗掉然後把COLOR DEMO 程式變成檔名字叫HELLO的一個檔。現在COLOR DEMO 程式就變成磁碟中的招呼程式了。爲了檢查所有工作是否如所預料的，BOOT DISK ，你將會得到像步驟2的程式。

你剛建立了－“轉動鎖”系統：每當在BOOT DISK時，它將自動的載入COLOR DEMO 程式並執行它。

鎖制及解鎖

有時你想防止一片磁碟被意外的清洗掉： LOCK 命令將爲你作這件事。

例子：打入

LOCK NESS， D2

這時磁碟內所含目錄會顯現一星號＊緊隣在 NESS 前方，

如果你覺得此檔不再需要鎖制，你可用UNLOCK 命令，使這

檔解鎖。

例子：打入

UNLOCK NESS

這些命令的語法如下：

LOCK f [, Ss] [, Dd] [, Vv]

UNLOCK f [, Ss] [, Dd] [, Vv]

記號的解釋請見第三章。

若你企圖去DELETE 或RENAME 一個已鎖制的檔時，你將得到下面的訊息：FILE LOCKED

如果你想 SAVE （保存）一個和鎖制檔同名字的檔，你也會得到同樣的訊息，（但是想保存的檔和鎖制檔需使用相同的語言才行。）

如果你想保存的檔，它和某一鎖制檔名字相同但所使用的語言不一樣，你將得到下面的訊息：

FILE TYPE MISMATCH

再試一次，並使用另外一個檔名字。

查證(verify)命令

偶而，磁碟沒有把資料正確的錄下來，譬如，如果磁碟有抓傷或骯髒物就可能發生這種事。而查證命令就是用來把這些受損或不正確寫入的檔報導出來的。它的語法和一般的 DOS 命令無異

VERIFY f [, Ss] [, Dd] [, Vv]

以下就是使用這種命令的例子：

VERIFY SAM

VERIFY FINANCE—8，D2，V22

VERIFY命令會先去檢查在指定檔上的情報是否相符，如果是的話，你得到的反應將是你所使用語言的催促信號，它們是：

＞整數BASIC型；

］Applesoft 型；

＊監督程式內時。

但是，VERIFY命令不會去檢查某一程式是否受損，如果在保存某一程式時損害已經發生，那麼在磁碟中它仍是損害的，而查證後的結果仍是無錯誤的。

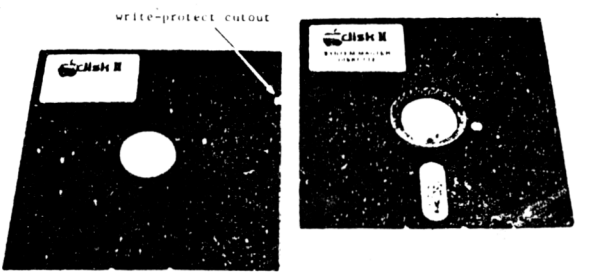
如果 VERIFY命令發現了錯誤，下面的訊息將出現：

I／O ERROR

如果你想查證一不在磁碟上的檔，將會得到如下的訊息：

FILE NOT FOUND

在整數BASIC，APPLESOFT，及監督程式中，你都可以使用VERIFY 命令。在這些語言中，你可查證任何型態的檔，例如主文檔（第6、7、8章）及機械語言程式（第9章）。



磁碟的禁寫入

LOCK 命令可以讓你保護一特別的檔，但有時候磁碟上所有的檔都希望被保護，不因外界的意外寫入而破壞資料。要達到此目的，你只需把磁碟邊上的方形缺口蓋上即可。當你買磁碟時這種蓋上缺口的黏貼標籤也會隨著附送，其它任何夠強韌的帶子也都可以勝任。注意，系統主磁碟並沒有禁寫入的缺口：它已經永遠的被禁寫入了。

如果你決定再重新讀寫一禁寫入的磁碟，只要把貼在缺口上的標籤移走即可。

有些程式不能存於禁寫入磁碟中，其中的一個例子就是 ANIMALS 程式，它是系統主磁碟內中的展示程式之一。把系統主磁碟放入磁碟機中，BOOT DOS（如果需要的話），打入

LOAD ANIMALS

這指令會把ANIMALS 程式放到主記憶體中去，再打入 RUN 後，螢光幕會出現

WRITE RROTECTED

STOPPED AT 1Ø4Ø

訊息。

ANIMALS 不能與禁寫入的磁碟共同使用，因爲它在執行時需把資料寫入磁碟上去。當你執行此程式時，在磁碟機上的磁碟必須是可寫入的，否則即無法執行。

現在，ANIMALS 程式雖然在主記憶體當中，但如果使用系統主磁碟的話就無法執行它。這時，放入一已啓始的且可寫入的磁碟到磁碟機中，然後打入 RUN ，現在你就可執行ANIMALS 程式了，它會把你“教”的事記在磁碟上，當你做完這程式後，打入

SAVE ANIMALS。此程式就被保存在此可寫入之磁碟之中了。

如果現在打入CATALOG 。那麼不但在磁碟上有ANIMALS程式的副本，而且還會有一由ANIMALS 程式所產生的一個檔叫做ANIMALSFILE 。

防止你遭受不幸的方法

與其它計算機儲存資料方法比較起來，軟性磁碟已可算是相當的可靠及耐用。但仍有可能因爲不小心而失去了磁碟上的資料，例如，磁碟可能被抓傷，被狗咬到，被人當飛盤丟，若不是禁寫入的磁碟它還可能被人蓋掉，還有，磁碟也會用壞－平均大約 40 個工作小時。

＊＊教　訓＊＊

一個程式最好保存一個以上的副本如果你不想因意外而失去它的話。用計算機術語來說：“back-up”（儲備）任何有價值的程式。

如果你在編寫或修改程式當中，back-up 程式的方法就是保留先前版本的副本，如果現在這一版不愼遺失的話，你還可找到以前最新的版本，這樣一來可使你的損失達於最小。有一種很好的方法就是在每一檔名字後面加上一個數字，當有修訂版時只要換掉後面的數字即可。舉例來說，假設你開始寫了一個程式叫作FINANCE，第一次保留的版本可稱爲FINANCE－1 ，下一次修改後版本就可稱爲FINANCE－2 ，第三次FINANCE－3 ，以次類推。在結束時你將有一堆FINANCE 程式，其中後面數字愈大版本愈新。

把一發展中的程式定期的保留下來將是非常好的主意，如果你

每15到20分鐘做一次的話，即使一個意外的斷電或其它事故都不能洗去所有你的作品。你能夠在保存好程式後立即繼續工作－只要記得在下次 SAVE 時指定－新的編號就可。如果磁碟接近了滿載，這時就可把最前面幾個版本刪除掉以騰出空間來裝新的。留幾個版本下來是很值得的，有時是爲防止意外發生，有時是你也需要以前的版本，因爲最新的版本不見得一定有所改進，說不定是大作虛功一場。

備用版本也可以是在不同磁碟上來存放不同的版本，有兩種方法：第一種，使用一磁碟機，把程式 SAVE 到在磁碟機內之磁碟，然後抽出此片再放另一片進去，再 SAVE 一次即可。第二種，乾脆把整片磁碟複製一片，下一節即是討論第二種方法。

複製 (COPY) 程式的使用法

你能夠使用 COPY 程式（它原先存於系統主磁碟上）來把你現今磁碟上所有的程式複製到備用的磁碟上去。如果你是以 APPLESOFT BASIC 程式與 DOS 相連，就需使用COPYA程式來複製磁碟。

在這些複製程式中，被抄的磁碟稱作“原版”（original）磁碟，原版磁碟的所有內含將被抄入－“複製”（duplicate)磁碟中，複製磁碟在複製前並不需被啓始。事實上，在複製時，原先存於複製磁碟的資料均將被洗掉。

在複製原版磁碟時，最好先使原版磁碟禁寫入；因爲這樣即使你放把原版和複製版放錯地方的話，也不會意外地把它的內容洗掉。

COPY 程式在沒有指定磁碟機的狀況下，就會自動假定原版磁碟位於現所設定溝槽控制板的磁碟機上，而複製版磁碟則位於上

述控制板所接的另一磁碟機上，當然以上所指的都是自然值，在螢光幕上首先會顯示出自然值，如果正是想使用這些自然值的話只要按下RETURN 鍵即可；若自然值有需更改之處，不論是對原版或複製版磁碟的所在而言，就必須再打入正確的數字來修改螢光幕上的自然值。以下就是一個例子，它是只以單一個磁碟機在APPLE上執行 COPY 程式。在此存放 COPY 程式磁碟的磁碟機和要抄寫的磁碟所在之磁碟機相同，接在一控制板的接線槽上。

1. 把系統主磁碟放到現今系統所指的磁碟機上。打入 RUN COPY或RUN COPYA (Applesoft 語言的話) 在一陣攪動之後，你應看到：

```
APPLE DISKETTE DUPLICATION PROGRAM

ORIGINAL SLOT:  DEFAULT = 6
```

2. 如果想使用程式的自然溝槽值，在我們例子中是 # 6 溝槽，只需按下RETURN 鍵即可。
3. 當 `DRIVE:  DEFAULT = 1` 訊息出現時，按下RETURN 鍵。
4. 當你看到

   `DUPLICATE SLOT:  DEFAULT = 6`

   再按下RETURN 鍵。
5. 當你看到複製版磁碟所在的磁碟機自然值時

   `DRIVE:  DEFAULT = 2`

   打入1 。
6. `-- PRESS 'RETURN' KEY TO BEGIN COPY --`

訊息將會出現，這信號通知你把系統主磁碟從磁碟機拿出，然後插入想複製的原版磁碟。（注意！你有沒有令你的原版磁碟禁被寫入。）

7. 現在 COPY 程式已在執行當中，它首先通知你插入你的原版磁碟，然後再告訴你它已在讀原版磁碟的資料，以後它會通知你何時把原版磁碟換成複製磁碟，及何時它正在啓始這片複製磁碟。在複製磁碟被啓始後你將被指示再把原版磁碟插入以使此程式能再讀入更多的資料，然後再插入複製磁碟以抄寫這些資料。上述步驟繼續反覆不斷執行直到所有原版程式全部被抄入複製磁碟爲止。

8. 當抄完後，你會看到下面的訊息：

```
DO YOU WISH TO MAKE ANOTHER COPY?
```

如果你打入Y（yes），系統將使用現今所使用原版磁碟及複製磁碟的位置，重覆以上的步驟。當你看到：

```
-- PRESS 'RETURN' KEY TO BEGIN COPY --
```

要記得再把你的原版磁碟插入，並再準備一空白或想重錄的磁碟當做複製磁碟。如果不想再複製的話，只需打入N即可；此時，你就離開了COPY 程式，而 DOS 假定此時的磁碟機自然值爲複製磁碟所在位置之值。

如果你有一台以上的磁碟機，你可在 COPY 程式執行之前把原版磁碟和複製磁碟所在的磁碟機指定好，然後把磁碟插入。注意，當你使用一台以上的磁碟機，系統就不會顯示信號催促你把磁碟插入或拔出，那種情況只在只有一台磁碟機時才會發生。

如果你想把資料寫入一禁寫入的磁碟中，將會得到：

I／O ERROR

STOPPED AT（停下那行的編號）

除非禁寫入的貼紙被取走，否則沒有任何資料能進入這片禁寫入的磁碟中。

如果磁碟機的門開著或裏面沒裝磁碟，I／O ERROR或UNABLE TO READ訊息將會出現。這些訊息也可能是指示磁碟機內之磁碟有問題。注意當你得到這訊息時，磁碟機自然值也許發生了改變。當你重新執行COPY 程式時，你可能需要再指定系統主磁碟所在的磁碟機位置。

第　五　章

更多的“管家”方法

除錯：監視及非監視命令
（MON及NOMON）
MAXFILES命令
追踪(TRACE)命令
如何使用MASTER CREATE程式

除錯：監視及非監視(MON及NOMON)命令

修改程式使其照著你所預計一般地執行的過程叫做除錯．程式中含有的錯誤通常稱爲“蟲”。所有磁碟命令及計算機和磁碟之間所互通的訊息通常都不在螢光幕上出現。但當你在除錯狀態當中，監視這些訊息能夠幫助你把問題追踪出來。

MON 命令能讓你監視許多的情報，如想再回到螢光幕上不顯示這些情報的狀態則可使用NOMON命令。

有三個參數可能出現在以上的命令當中：

C　表示給磁碟的命令（如OPEN，READ 等等）

I　表示從磁碟輸入（當讀入一檔時）

O　表示輸出至磁碟（當寫出一檔時）

這些參數只能與NOMON及 MON 命令合用。通常時候，系統在NOMON C、I、O狀態：代表沒有任何監督狀態。

上述命令的語法如下：

MON [C] [,I][,O]

NOMON [C] [,I][,O]

這兩種命令至少需要有一個參數，不然就不被系統接受。參數間需以逗號分開，可以任意秩序排列。

有7種不同的方法來使用 MON 命令：

命　令	所監督之事
MON C	到磁碟的命令
MON I	從磁碟來的輸入
MON O	到磁碟的輸出
MON I,O	磁碟的輸出與輸入
MON C,I	到磁碟的命令與自磁碟的輸入
MON C,O	到磁碟的命令與至磁碟的輸出
MON C,I,O	到磁碟的命令與磁碟的輸出與輸入

＊　＊　＊　注　　意　＊　＊　＊

一 MON 命令在任何情況下永遠有效，除非遇到了①NOMON，INT，或FP（只在靭體狀況下）等命令爲止。

或　　者

② Boot 系統時

或　　者

③ 執行 restart 程式（3DØG）

或　　者

④ 重置程式時

一個俐落的方法：你可先發布一 MON 命令稍後再取消它而不會影響到螢光幕上的格式——即使NOMON並未顯示在螢光幕上。假定你執行了某一種 MON 命令，譬如 MON C，I，O爲了要取消這命令並使它不顯示在螢光幕上，可打入

PRINT D$;″NOMON C,I,O″:VTAB PEEK(37):CALL-868

其中D$ 和以前一樣代表CTRL—D。

MAXFILES命令

DOS 系統允許 16 個檔同時使用。DOS 除了到現在爲止所介紹的BASIC程式外還能夠處理許多型態的檔。參考第六章所介紹循順序主文檔，第八章所介紹的亂序隨機存取檔及第九章 DOS 命令用來處理二進位檔。

MAXFILES命令指定有多少檔可以同時使用的數目。當你 BOOT DOS 後，MAXFILES 3 命令就被執行。此命令建立了一自然值狀態：在還沒有遇到 MAXFILES 命令之前，最多有

3個檔能夠同時執行。

此命令的語法如下：

MAXFILE n

n必須是1到16間的整數，超過了此範圍，如在APPLE-SOFT 或整數BASIC程式當中，系統就會產生 SYNTAX ERROR的信號；若在監督程式狀態時，系統只會“嗶”一聲告訴你犯了某一種錯誤。

對每一指定的檔，MAXFILES 命令會指定595拜的記憶空間（叫做檔緩衝區）。這些另外加的記憶空間是用來幫助調整取用記憶體資料及取用磁碟資料時速度的差異（取用磁碟資料時速度較慢因爲需要機械動作）。因此，在效率的要求下，一檔緩衝區就被使用來暫時儲存要到磁碟去或從磁碟來的資料。

如果你想從磁碟上取用資料時，DOS 一次搬進256個字元，把它們放入檔緩衝區的輸入部份，然後從這256個字元內找出你的程式所要的資料。如果你想把資料送到磁碟，所要送的字元也會暫時存在檔緩衝區的輸出部份，直到滿了256字元後一次輸送到磁碟上去。

假定你打入了MAXFILES 1命令，因此只有一個檔可使用，如你想使用其它 DOS 命令（例如CATALOG）系統會告訴你 NO BUFFER AVAILABLE 。因爲執行這些命令需用兩個檔以上才可。

當 BOOT 系統時，“使用中”檔的數目自然值爲3，所以需1785 拜的記憶空間來當作檔緩衝區。在大多數的情況下，你不會需要3個以上的檔同時使用。如果實在需要更多的檔，可在載入與執行程式之前打入 MAXFILES n（其中n是所需要檔的數目）。

如果程式執行中才更改 MAXFILES 的話將可能洗去整數BASIC程式，並使APPLESOFT字串和MAXFILES發生混亂的機會增加。因爲最高使用記憶體指標被下降，但是程式或字串卻沒有。爲了要避免這問題，還是在載入及執行程式之前指定好 MAXFILES 爲良策。

如果 MAXFILES 命令使用在程式當中，將會改變記憶體指標，如此，GOTO及GOSUB 及其它指令將指錯地方。如果你實在必須在APPLESOFT程式中改變 MAXFILES ，那麼就把 MAXFILES 命令放在程式的第一行（在宣告任何變數之前），例如，10 PRINT CHR$(4);"MAXFILES 5":：

如想在整數BASIC程式中使用 MAXFILES 時，你就必須先創出 EXEC 檔，詳細討論在第七章前部。

追蹤（TRACE）命令

在APPLESOFT中TRACE命令是一很有用的除錯工具。但是當使用TRACE命令時，在APPLESOFT程式內的 DOS 命令就失靈了，這是因爲TRACE命令會在 DOS 命令前印出行編號來。有個解決部份問題的方法，可把一個 RETURN（即CHR$(13)）字元插入到D$ 後的字串中：譬如 10 D$ =CHR$(13)+CHR$(4)。這樣，即使在使用TRACE命令的情形下，大多數的 DOS 命令仍能工作，但有一例外如下：

如果TRACE命令在使用中而 DOS 如想從磁碟中讀入一筆資料，但此時程式將停下來到鍵盤等輸入。在與 DOS 命令有關時，

TRACE命令的用途將因此受到限制。

如何使用MASTER CREATE 程式

如第二章所討論，INIT 命令是用來產生奴磁碟。在本節中，你將學到如何來產生主磁碟。主磁碟及奴磁碟外形上的差別並不是很明顯的：它們都是以黑塑膠製成的。要分辨它們需要靠你不斷的去修正招呼程式及磁碟標籤以資識別。

系統主磁碟中包含有一程式叫作MASTER CREATE（主磁碟創建）程式，它需在至少 16 K記憶體的APPLE Ⅱ機裏執行，MASTER CREBTE程式能爲你做下列事：

* 把一奴磁碟（其 DOS 和記憶體大小有關）轉變成主磁碟（其 DOS 則可自行重定址，因此記憶體在任何大小系統下能有效運用）。
* 可以給轉換過的磁碟一個新的招呼程式名稱，每當 BOOT 磁碟時，DOS 就會去執行這名字所代表的程式。

MASTER CREATE程式必須與已被啓始的磁碟合用且此磁碟不得是禁寫入型的。

以下是一個例子，用來顯示如何提升如第二章所述已啓始好的磁碟（ONE TO TEN 程式所在的那片）成爲一片主磁碟。爲方便起見，在以下的討論中這片磁碟將被稱作是 ONE 磁碟。

在使用MASTER CREATE之前，先做下列事：

1. 把你想轉換的磁碟——在我們的例子中就是 ONE 磁碟——插入磁碟機內，然後執行這片磁碟的招呼程式——在ONE磁碟上名叫HELLO。招呼程式所顯示的訊息必須包括啓始這片磁碟所用的 DOS 版本，及它是主或奴磁碟的狀態。

2. 修改招呼程式使它能夠顯示新的情報“MASTER DISKETTE”。然後把改好的招呼程式保存（SAVE）下來。磁碟外所貼的標籤若需要更改的話，現在就更正它。
3. 注意招呼程式的名字。如果想在每次 BOOT 磁碟時都執行和以前相同名字的招呼程式的話，在以後可給MASTER CREATE 程式相同的招呼程式名稱。如果希望有不同名字的招呼程式，也可RENAME（重新命名）招呼程式，在稍後應通知MASTER CREATE程式新的招呼程式名字。

要使用MASTER CREATE程式時，可照下所述去作：

4. 把系統主磁碟放入磁碟機內，BOOT DOS；不管是使用兩種BASIC程式中的那一種，打入：

   BRUN MASTER CREATE

5. 之後，你將看到下面訊息：

```
DOS 3.3 MASTER-CREATE UTILITY
COPYRIGHT 1980 BY APPLE COMPUTER INC.
ALL RIGHTS RESERVED

(NOW LOADING DOS IMAGE)
```

6. 螢光幕會通知你輸入招呼程式的名字給轉換後的磁碟使用

```
PLEASE INPUT THE GREETING PROGRAM'S
FILE NAME:
```

我們將假定修訂後的招呼程式仍然叫做HELLO，所以現在打入：HELLO。

若你用不同的名稱則在下次BOOT DISK時需注意要記得起這名稱。當你按下RETURN 鍵就進入了招呼程式內

，此時將看到：

```
REMEMBER THAT MASTER DOES NOT CREATE
THE GREETING PROGRAM, OR PLACE IT IN
THE DISK DIRECTORY
  THIS IS THE FILE NAME THAT WILL BE
PLACED WITHIN THE IMAGE:
      HELLO
  PLACE THE DISKETTE TO BE MASTERED IN
THE DISK DRIVE.
  PRESS <RETURN> WHEN READY
NOTE: IF YOU WANT A DIFFERENT FILE NAME,
PRESS <ESC>
```

7. 照著指示去做。從磁碟機把系統主磁碟取出，把想轉換的磁碟放入磁碟機－在此例中即磁碟 ONE ，最後，按下 RETURN 鍵開始轉換；程式將通知你何時此步驟會完畢。
8. 使用MASTER CREATE程式後，在做任何其它工作前永遠要記住要重新BOOT DOS。

* * * 注 意 * * *

你給MASTER CREATE程式的名字（即招呼程式名字）並不會放在磁碟的檔案目錄(CATALOG)內，它只是通知磁碟的DOS當在 BOOT 磁碟時須執行的招呼程式名字是什麼。你必須確定磁碟的檔案目錄內是否含有和招呼程式名字相同的程式名稱。

如果你忘了做上述步驟（即跳過上面第 3 步驟）那麼每當你用此磁碟做 BOOT 動作時，你將看到：FILE NOT FOUND

* * 提 醒 * *

你必須記得每片磁碟招呼程式的名字才行。最好的辦法是所有的磁碟都使用相同的招呼程式名字。

第　六　章

如何使用循序主文檔

主文檔：簡介
循序主文檔：一些例子
開啓及關閉循序檔（OPEN　CLOSE）
循序檔的寫入（WRITE）
循序檔的讀出（READ）
更多有關循序檔資料：APPEND及
POSITION命令

拜定位後處理資料的方法。

主文檔：簡介

有時，你想用磁碟來儲存一些並不是程式的資料，譬如，也許你希望有一份通訊錄的副本、有關現猜字遊戲的一些字串、一計算的中間結果，或是寄信地址目錄等等。一個主文檔，有時也叫做資料檔，可以幫你處理這些以及其它更多的事情。在目錄索引（CATALOG directory）中，T字母開頭的都代表主文檔。

主文檔在整數BASIC及APPLESOFT程式中是以DOS 命令產生及取用的。一主文檔能夠由其中一種語言寫成的程式所創出然後被另一種語言所寫成的程式所取用。

大多數的程式例子是以APPLESOFT語言所寫成。如果你想轉變成整數BASIC 型時，記住在整數BASIC語言中不能使用字串列（string array），必須宣告行列（DIMention）字串變數。在一些整數BASIC 命令中，如

INPUT A$，B$，C$

只有RETURNs （而不是逗點）可以分開這3種反應。這本手冊並不打算告訴你如何使每一程式以整數BASIC語言執行：在APPLESOFT· Ⅱ BASIC程式參改指引，附錄M內有更詳細有關程式轉換的說明。至於程式寫好後如何改變程式型態來執行請見本手冊下一章所述。

DOS 內的命令LOAD及RUN （或BLOAD及BRUN）不能使用於主文檔中，如果你如此做，將會有FILE MISMATCH的訊息出現。LOAD及RUN 命令只能在BASIC程式檔用（正如BLOAD 及BRUN 只能在二進位制機械語言中使用一般），在主文檔中就不行。如果你寫了一些程式，這些程式必須傳送資料到主文檔或從主文檔中取出資料，你就必須使用本章所討論的DOS 命

令如下：

OPEN

CLOSE

READ

WRITE

APPEND

POSITION

EXEC

其中OPEN，CLOSE，APPEND 及POSITION 並不能使用於立即執行模式。如果你如此做，你將得到NOT DIRECT COMMAND 。這些命令必須以延遲執行模式（deferred-execution mode） 執行，也就是說，在執行某一程式中間接執行這些命令。至於CLOSE及EXEC命令則可以立即執行模式執行。

除了上述那些命令，其它DOS 命令如：

LOCK 及UNLOCK

DELETE

RENAME

MON 及NOMON

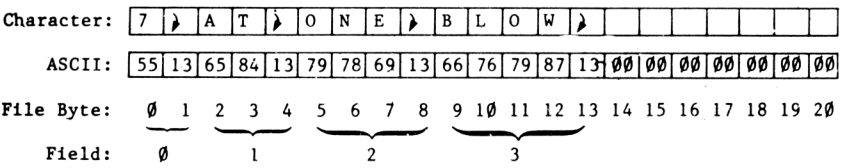
VERIFY

CATALOG

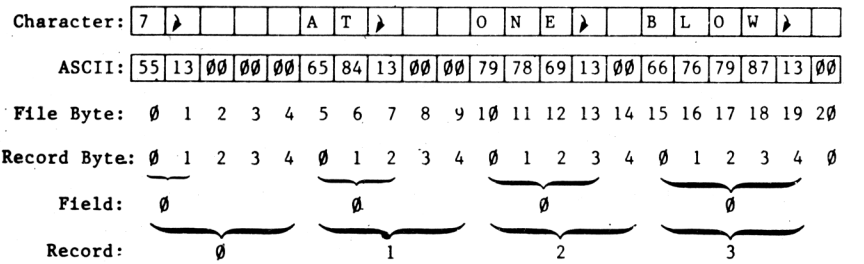
它們在主文檔上的使用法和程式檔中一模一樣。

有兩種不同型式的主文檔：循序主文檔及隨機存取主文檔。兩種主文檔的資料都以ASCII 碼方式儲存，但有不同的格式，這兩種主文檔型式的圖示如下（↵ 代表RETURN 字，在大多數PRINT 敍述之後都會自動送出）。

循序主文檔



隨機存取主文檔（例子：資料錄長5拜，每錄一欄）



欄（field）及資料錄（record）將在6，7，8章討論。OPEN，CLOSE，READ，WRITE及POSITION這些命令在這兩種不同型式主文檔中都可使用。但使用法稍有不同。循序主文檔在某些方面較易使用及了解，所以我們首先討論循序主文檔的用法及構造。隨機存取主文檔在第8章再予說明，有關所有各種不同型式的檔的技術資料可在附錄C中查得到。

循序主文檔：一些例子

在猜字遊戲中，假如你想建一檔，此檔包含所需用字的一覽表。在此有兩對程式來做此事。每一對的第一個程式在磁碟上產生一主文檔，第二個程式則從此磁碟上取用該項資料。

下面的程式產生一一名叫WORDS1 的文檔，它包含有APPLE，BANANA ，CATALOG ，DORMANT，EAGLE，FRUIT，GOOSE，HAT 及ICICLE 等字。

```
10  REM   MAKE WORDS1
20 D$ = "": REM   CTRL-D
30  PRINT D$;"OPEN WORDS1"
40  PRINT D$;"WRITE WORDS1"
50  PRINT "APPLE"
60  PRINT "BANANA"
70  PRINT "CATALOG"
80  PRINT "DORMANT"
90  PRINT "EAGLE"
100  PRINT "FRUIT"
110  PRINT "GOOSE"
120  PRINT "HAT"
130  PRINT "ICICLE"
140  PRINT D$;"CLOSE WORDS1"
150  END
```

在30 行開啓（OPEN）此檔，它使用了在BASIC程式中傳送命令給DOS 的標準規格。OPEN命令在檔目錄（CATALOG）內放進了一名叫WORDS1 的主文檔（如果目錄裏先前並無此名字的話）。

編號40 行的WRITE命令使得以後的PRINT指令所輸出的資料往以上定名的主文檔送而不往電視螢光幕上送。因此，在此程式裏，從編號50 至130行的PRINT 命令將會把在引號內的字送到主文檔WORDS1 內去而不是送到電視螢光幕。

編號140行的指令關閉（CLOSE）此檔，結束了寫入的過程。

如果此程式被執行且又不在監督情況（MON）下，在螢光幕上你將看不到任何動靜：經常DOS 命令以及輸出入磁碟的動作都不會顯現出來。但是，如照第 5 章所述，打入

MON C，I，O（或者簡單點MON C，O，因爲此例中沒有從磁碟輸入資料）。執行上面程式後你將會看到：

```
OPEN WORDS1
WRITE WORDS1
APPLE
BANANA
CATALOG
DORMANT
EAGLE
FRUIT
GOOSE
HAT
ICICLE
CLOSE WORDS1
```

到此，你在磁碟上存入了一叫WORDS1 的檔，在目錄上WORDS1 檔將被標示爲T表示它是主文檔。這檔包含有以RETURN隔開的各項資料（在本例中爲各個字〔word〕）。如果在PRINT指令後沒有逗點（，）或半分號（；），那麼在PRINT指令後就會自動加入一RETURN 字元。注意，在此RETURN只是一個字元（character）而非動作（action）一它只是ASCII碼13 ，如此而已。

每個以RETURN 字元當結尾的資料項叫做欄（field），一欄是由一列ASCII編碼的字元所組成，而最後的字元必須是RETURN，以上資料皆存於主文檔內。

WORDS1 檔就是一循序主文檔（Sequential text file），它的每一欄的起始及終了都是RETURN 字。欄可以有不同的長度：如APPLE佔了6 拜的位置（5 個字母佔了5 個拜，RETURN佔了一拜），BANANA 佔了7 拜，以次類推。循序主文檔是以一長而連續ASCII編碼的字元串成存於磁碟中的，而欄與欄中沒有留間隙。

好，現在WORDS1 已在磁碟中了，再來的一個問題“怎麼把它取出來用”？，以下的APPLESOFT程式就可以做這件事：

```
10  REM   RETRIEVE WORDS1
20 D$ = "": REM   CTRL-D
30  PRINT D$;"OPEN WORDS1"
40  PRINT D$;"READ WORDS1"
50  FOR I = 1 TO 9
60  INPUT A$(I)
70  NEXT I
80  PRINT D$;"CLOSE WORDS1"
90  END
```

編號30 行開啓（OPEN）此檔；編號40 行指令告訴DOS 以後所有的INPUT或GET 指令都是對才開啓的檔而言而不必要到鍵盤去找資料。事情就像是由磁碟打入資料一般，而不是你自己。一INPUT命令永遠是對一完整的欄整個“鍵入”APPLE主機而言。如果另一INPUT命令接著來，那將使下一個欄被讀入。從50 行到70 行就是使DOS 去讀入9 個欄分別放在列陣A$(1)，A$(2)，A$(3)……A$(9)中，80 行關閉（CLOSE）此檔。

當在執行以上程式時若無使用MON C，I，O命令時，在螢光幕上你將看不到任何東西。但若使用的話，則將看到：

```
OPEN WORDS1
READ WORDS1
?APPLE
?BANANA
?CATALOG
?DORMANT
?EAGLE
?FRUIT
?GOOSE
?HAT
?ICICLE
CLOSE WORDS1
```

從磁碟內輸入APPLE的資料前面都加了一疑問號（？）。

現在檢查是不是如上所說的，打入

PRINT A$(2), A$(9), A$(4)

你將首先看到BANANA ——從A$(2)而來，然後ICICLE及DORMANT。這是檢查資料是否正確讀入的一個很好方法。

如你修改了MAKE WORDS1 程式，使它造的字和原先的不同，注意在還沒有重新執行MAKE WORDS 1 前須先DELETE WORDS 1，如果忘了話，得到的結果將是新字與舊字混合在一起。

以下展示如何來創出一名叫WORDS 2 的循序檔，它所包含的字和WORDS 1 相同，但九個字都在一欄內。每一字後面都接了一個逗號，因此一INPUT 命令帶著許多變數（在此例中有9 個）能被用來取出各別的字。

```
10  REM     MAKE WORDS2
20 D$ = "": REM   CTRL-D
30  PRINT D$;"OPEN WORDS2"
40  PRINT D$;"WRITE WORDS2"
50  PRINT " APPLE, BANANA, CATALO
     G,";
60  PRINT " DORMANT, EAGLE, FRUIT
     ,";
70  PRINT " GOOSE, HAT, ICICLE"
80  PRINT D$;"CLOSE WORDS2"
90  END
```

注意，50 行的PRINT命令是以半分號做結束的。如果命令後面有個半分號，那麼RETURN 字元就不會再自動地加入該命令之後。因此，下一個PRINT命令所送出到磁碟的字將會和50 行PRINT命令所送的字同一欄，接著60 行PRINT命令後面又是一半分號，因此，該欄仍然還沒有一終止字元RETURN 出現，現

在70 行PRINT命令後並沒有逗號或半分號，使得一PETURN字自動加入而結束了此欄。所以，此欄包含了50 行，60 行及70 行所要印的字。

在沒有磁碟的情況下，連在PRINT命令後的逗號經常是代表在影幕上要送出的標籤欄（Tab-fields）字元；但是，在寫資料到磁碟上去時，PRINT 命令後的逗號就不同了。這些逗號和半分號的功能相同。在印（PRINT）到磁碟的動作中，由逗號分開的項目就會被接串（concatenated）起來，項目之間不留空隙。在PRINT 命令後的逗點和半分號的效果相同：項目終了時若遇到這些符號系統都不會自動送RETURN 字元上去。

當MAKE WORDS 2 正執行當中且在MON C，I，O監督狀態下，你將看到

```
OPEN WORDS2
WRITE WORDS2
 APPLE, BANANA, CATALOG, DORMANT, EAGLE,
 FRUIT, GOOSE, HAT, ICICLE
CLOSE WORDS2
```

下面這APPLESOFT程式取用WORDS 2 檔：

```
10  REM     RETRIEVE WORDS2
20 D$ = "": REM    CTRL-D
30  PRINT D$;"OPEN WORDS2"
40  PRINT D$;"READ WORDS2"
50  INPUT A1$,A2$,A3$,A4$,A5$,A6$
     ,A7$,A8$,A9$
80  PRINT D$;"CLOSE WORDS2"
90  END
```

當上面程式在MON C，I，O監督狀態下執行，你將看到

```
OPEN WORDS2
READ WORDS2
? APPLE, BANANA, CATALOG, DORMANT, EAGLE
, FRUIT, GOOSE, HAT, ICICLE
CLOSE WORDS2
```

在整數BASIC 型程式內，對數值變數（字串變數則不適用）逗號能夠分隔多重輸入回應（multiple INPUT responses）。對字串變數只有和INPUT 命令配合以PETURN 字元來分隔多重反應。因此，在整數BASIC 型程式內，RETRIEVE WORD2 將會把整個欄（包括9個字，8個逗號，及6個空白）指定給變數A1$，當沒有任何欄可指定給A2$ 時，你將得到END OF DA DATA 訊息。

在APPLESOFT BASIC 內，你也可以使用GET 命令來從主文檔中取出資料（以一個字元一個字元的方式）。這有下列的優點：例如，能夠定義任何字元來標示字的結束（end-of-word），以下APPLESOFT程式也可從主文檔WORDS 2 取出資料：

```
10  CLEAR : REM  GET WORDS2
20 D$ =  CHR$ (4): REM  CTRL-D
30 R$ =  CHR$ (13): REM  RETURN
40 T$ =  CHR$ (1): REM  CTRL-A
50  PRINT D$;"OPEN WORDS2"
60  PRINT D$;"READ WORDS2"
70 I = I + 1
80  GET B$
90  IF B$ = "," THEN  GOTO 70
100  IF B$ = R$ THEN  GOTO 130
110 A$(I) = A$(I) + B$
115  PRINT T$;A$(I)
120  GOTO 80
130  PRINT R$;D$;"CLOSE WORDS2"
140  END
```

在第10 行，CLEAR命令使所有的變數（包括I 及所有的A$(I) 置爲0，第20 行使用APPLESOFT裏的另一種方法來把D$指定爲CTRL-D （4是CTRL-D的ASCII 碼）。這方

法使控制信號能看得見，印得出。

第 80 行GET 命令從主文檔WORDS 2 一次一字元的拿出資料，此檔在第50 行及60 行被開啓（ OPEN）及讀出（ READ），如果讀出的字元不是一個逗號也不是個RETURN 第110行就會把此讀出的字元加到A$(I)的尾巴去，第 120行把程式指標送回第 80 行去再拿（ GET）下一個字元。於是，此程式一個字元一個字元的把A$(1)所代表的第一字（word）凑出來了。

當拿出的字元是逗號時，代表第一字已結束，所以第90行把程式定位指標送回到第 70 行，使 I 增加 1 ，此表示要取第 2 個字，然後再開始一個字元一個字元地收集起來放到A$(2) 中去，以次類推。到最後，當碰到RETURN 字（R$）時，它代表此欄的結束，所以第 100行把程式定位指標送到130行來關閉（ CLOSE） 此檔及結束此程式。注意，在第 30 行所使用到的CHR$(13)，因爲在程式鍵入當中，如果打入 RETURN ，在螢 光幕上就會自動換行，所以要以代號取代它，CHR$(13) 在APPLESOFT中即代表 RETURN 。

當GET 命令從磁碟中拿出字元時，這些字元即使在MON C，I ，O（監督狀態）下亦不顯現於螢光幕上，第 115就是爲了顯現字是如何一個字元一個字元地建出來而設的。

當此程式在MON C，I ，O的情況下被執行，你可看到下面（應成一“長”行，而非圖上所示的 3 行）：

```
OPEN WORDS2
READ WORDS2

A                D                G
AP               DO               GO
APP              DOR              GOO
APPL             DORM             GOOS
APPLE            DORMA            GOOSE
                 DORMAN
B                DORMANT          H
BA                                HA
BAN              E                HAT
BANA             EA
BANAN            EAG              I
BANANA           EAGL             IC
                 EAGLE            ICI
C                                 ICIC
CA               F                ICICL
CAT              FR               ICICLE
CATA             FRU              CLOSE WORDS2
CATAL            FRUI
CATALO           FRUIT
CATALOG
```

在APPLESOFT型程式裏GET 命令從磁碟主文檔得到反應之後，下列幾個問題將發生：

1. 在NOMON C，I，O狀態下GET 命令後所要印的第一個字元將不會出現在螢光幕上。
2. 在MON C，I，O狀態下GET 命令後所要印的第一個字元將出現在螢光幕上。
3. 在上述兩種模式中，如果一DOS 命令是在GET 之後第一項要印出的，此DOS 命令可能不被執行，因爲它前面所需要的RETURN不見了。

在GET WORDS2 程式裏第115行，那沒在螢光幕上出現“可丟棄”字元CTRL-A（即T$）是放在想要印出的第一個字元之前，這招解決了上述問題1與問題2。爲了解決問題3，在第

13Ø行中可看到一個RETURN（即R$）被放到想要印出的DOS命令之前，其情況和TRACE命令一般。

現在來看最後一個例子，以下是一APPLESOFT程式，它產生一叫WORDS 3 的檔，其中第一欄有 2 個字，第 2 欄有 3 個字，第 3 欄有 4 個字：

```
10  REM  MAKE WORDS3
20 D$ =  CHR$ (4): REM CTRL-D
30  PRINT D$;"OPEN WORDS3"
40  PRINT D$;"WRITE WORDS3"
50  PRINT "APPLE,BANANA"
60  PRINT "CATALOG,DORMANT,EAGLE"

70  PRINT "FRUIT,GOOSE,HAT,ICICLE
    "
80  PRINT D$;"CLOSE WORDS3"
90  END
```

第一欄資料爲：APPLE，BANANA 爲13 拜長（每一字元一個拜再加一RETURN 字母）。第二欄的資料爲：CATALOG，DORMANT， EAGLE爲22 拜長；第三欄資料爲：FRLIT，GOOSE，HAT，ICICLE 爲23 拜長。在MON C，I，O之情況下執行此程式所得結果如下：

```
OPEN WORDS3
WRITE WORDS3
APPLE,BANANA
CATALOG,DORMANT,EAGLE
FRUIT,GOOSE,HAT,ICICLE
CLOSE WORDS3
```

以下的程式可取出WORDS 3的資料：

```
10  REM  RETRIEVE WORDS3:A
20 D$ =  CHR$ (4): REM  CTRL-D
30  PRINT D$;"OPEN WORDS3"
40  PRINT D$;"READ WORDS3"
50  INPUT R$,S$
60  INPUT T$,U$,V$
70  INPUT W$,X$,Y$,Z$
80  PRINT D$;"CLOSE WORDS3"
90  END
```

在MON C，I，O情況下執行上述程式，你將看到：

```
OPEN WORDS3
READ WORDS3
?APPLE, BANANA
?CATALOG, DORMANT, EAGLE
?FRUIT, GOOSE, HAT, ICICLE
CLOSE WORDS3
```

要讀出循序主文檔WORDS 1，WORDS 2，WORDS 3的程式必須要小心的設計來讀出欄的正確編號及欄中每項的正確編號。一般來說，要取用一主文檔的程式必須要依此檔的特殊形態而設計。如果弄錯的話，所得到的結果將會使人迷惑。舉個例子，考慮以下“不正確”的程式，它的目的是要取出主文檔WORDS 3的資料：

```
10  REM  RETRIEVE WORDS3:B
20 D$ = "": REM  CTRL-D
30  PRINT D$;"OPEN WORDS3"
40  PRINT D$;"READ WORDS3"
50  INPUT R$, S$
60  INPUT T$, U$, V$
70  INPUT W$, X$, Y$
80  PRINT D$;"CLOSE WORDS3"
90  END
```

在MON C，I，O的狀態下，執行上面程式後螢光幕有如下的結果：

```
OPEN WORDS3
READ WORDS3
?APPLE, BANANA
?CATALOG, DORMANT, EAGLE
?FRUIT, GOOSE, HAT, ICICLE
?EXTRA IGNORED
CLOSE WORDS3
```

在第70行的INPUT命令使包含FRUIT，GOOSE，HAT，ICICLE資料的欄讀入APPLE中，首先三個字被指定到

W$ ，X$ 及Y$ 三個變數中，但是第 4 個字 ICICLE 就找不到變數來對應了，因此有 EXTRA IGNORED 的訊息顯現，但程式是繼續執行。

以下是另一“不正確”程式，它也是從 WORDS 3 主文檔裏讀出資料：

```
10  REM  RETRIEVE WORDS3:C
20 D$ = "": REM  CTRL-D
30  PRINT D$;"OPEN WORDS3"
40  PRINT D$;"READ WORDS3"
50  INPUT R$, S$
60  INPUT T$, U$, V$, W$
70  INPUT X$, Y$, Z$
80  PRINT D$;"CLOSE WORDS3"
90  END
```

在 MON C，I，O 狀態下執行此程式結果如下：

```
OPEN WORDS3
READ WORDS3
?APPLE, BANANA
?CATALOG, DORMANT, EAGLE
??FRUIT, GOOSE, HAT, ICICLE
?EXTRA IGNORED
?
END OF DATA
BREAK IN 70
```

這次，第 60 行本打算使 CATALOG，DORMANT，EAGLE 讀入 APPLE 機中，此三字被指定到變數 T$，U$，V$ 名下，但第 60 行的 INPUT 命令希望有第 4 個對應，因此它把下一個完整的欄讀入 APPLE 機中：FRUIT，GOOSE，HAT，ICICLE 。這欄第一個字 FRUIT 被指定到第 60 行變數 W$ 名下。此時，INPUT 命令的變數用完了，下面 3 個字找不到對應的變數，因此 EXTRA IGNORED 的訊息就出現了。程式繼續執行下去。但是，在檔內再也找不到任何欄了。因此第 70 行的 INPUT 命令使

END OF DATA 訊息出現，這程式也到此結束。

一對稍微一般化的程式對：MAKE TEXT 及RETRIEVE TEXT將在稍後幾節討論。它們顯示出如何使一程式更能適應各種不同的主文檔。

開啓及關閉循序檔

（OPENing and CLOSEing SEQUENTIAL FILES）

循序主文檔應被用在當資料以一種線性方式從檔的開頭開始取出一直到結束，而且不需要經常的更新或更改版本，例如，一循序檔能被用來爲猜字遊戲儲存資料，就像前幾個例子一般。

爲了要產生一循序主文檔，需照著順序（但不必緊接著）使用以下命令：

OPEN
WRITE
PRINT
CLOSE

如要取用一循序主文檔的資料，則需照著順序（也不必緊接著）使用以下的命令

OPEN
READ
INPUT
CLOSE

此兩步驟皆已於上一節說明過。

在產生（或寫入）一循序主文檔之前或後有一些程序是要被遵守的：在使用這種檔前需要先開啓（OPEN）它；使用完後需要關

閉（CLOSE） 它。同理在取出（或讀出）這種檔時，在讀之前需先OPEN此檔，讀之後需CLOSE此檔。

檔一旦被開啓在以後一定要記得關閉。忘記把開啓的檔關閉若被誤寫進去就會使資料受損。

這些命令的語法和DOS 其它的命令相類似。〔注意：OPEN及CLOSE命令也適用於隨機存取檔——見第8章。〕

OPEN f〔, Ss〕 〔, Dd〕〔, Vv〕

CLOSE〔f〕

例子： OPEN SESAME

OPEN SHOP , D2 , S7

CLOSE MOUTHED

CLOSE WINDOW

OPEN在APPLE主機內會爲檔 f 儲存一些工作區（OPEN會指定一595拜的檔緩衝區來處理檔的輸入及輸出），並且使系統準備妥當開始從檔的開頭讀或寫。OPEN可指定以後WRITE（or READ）命令所需要的溝槽及磁碟機編號。

CLOSE命令把工作區歸還給APPLE主機。如果 f 沒有被指定的話，除了執行EXEC命令的檔以外，其它所有在OPEN 狀態中的檔都會被CLOSE掉。EXEC檔留到第7章再討論。OPEN有時也執行CLOSE動作：OPEN首先檢查看看名稱檔（named file）是否已經OPEN；如果是的話，它就在名稱檔重新開啓前把它關閉。

注意，CLOSE命令中沒有磁碟機及溝槽等參數。如果你打入：CLOSE MYFILE。那麼任何名叫MYFILE的檔通通都會被關閉，不管該檔是位於那個溝槽何架磁碟機上。同樣的單單一個

CLOSE將會關閉所有磁碟機上的檔（但除了正被EXEC命令支配的檔）。在許多情況下，你可能希望刪除一可能存在或可能不存在名字叫做 f 的檔。譬如在避免蓋寫不滿原先存在的檔(除非新檔能把原先存在的檔全部寫滿，否則未寫滿部份將會附著在新檔後面）的問題上就特別重要了。假定有一遊戲，每當它被玩時都需產生及使用SCORES 檔，現在希望在每個遊戲開始之前程式能刪除任何以SCORE爲名的舊檔。

若SCORES 檔不存在的話，DELETE SCORES命令將使FILE NOT FOUND 的訊息產生。下面是一快速的方法來消除任何名叫 SCORES 的檔並對新資料重新開啓它（不論此檔現在是否存在：

```
5  REM  SCORES DELETER
10 D$ = "": REM  D$ IS CTRL-D
15  PRINT D$;"OPEN SCORES"
20  PRINT D$;"DELETE SCORES"
25  PRINT D$;"OPEN SCORES"
30  REM
        REMAINDER OF PROGRAM
                HERE
```

循序檔的寫入

以下要介紹另外一程式，它也能產生一循序主文檔。此APPLESOFT 程式產生一叫 SAMPLE 的主文檔，此檔包含有3個字串及1Ø 個數字。

當此程式執行之時，SAMPLE 檔可能已經存在或不存在：如果它存在的話，它就會被刪除（ DELETE ）而使舊資料從檔中移去；如果它不存在而你的程式想要去刪除它，你將得到FILE

NOT FOUND的訊息，此程式也會停止。第 2Ø 行及第 3Ø 行處理此問題。如果SAMPLE檔已經存在，第 2Ø 行開啓它然後第3Ø行刪除它。而第 4Ø 行被執行後就會產生一清潔的新SAMPLE 檔，把資料弄混的問題就可避免了。

```
5  REM  MAKE SAMPLE
10 D$ =  CHR$ (4): REM. CTRL-D
20  PRINT D$;"OPEN SAMPLE"
30  PRINT D$;"DELETE SAMPLE"
40  PRINT D$;"OPEN SAMPLE"
50  PRINT D$;"WRITE SAMPLE"
60  PRINT "HI HO": PRINT "HI HO"
70  PRINT "OFF TO THE DISK WE GO"
80  FOR J = 1 TO 10
90  PRINT J
100  NEXT J
110  PRINT D$;"CLOSE SAMPLE"
120  END
```

在MON C，I，O狀態下執行此程式，在螢光幕上將看到

```
OPEN SAMPLE
DELETE SAMPLE
OPEN SAMPLE
WRITE SAMPLE
HI HO
HI HO
OFF TO THE DISK WE GO
1
2
3
4
5
6
7
8
9
10
CLOSE SAMPLE
```

在你寫入（WRITE）一一檔前，必須要把檔先開啓；工作完後也要輕輕的關掉。開啓及寫入的命令必須附有相同的檔名字。

寫入命令被執行後，任何接著而來的印出（PRINT）命令會把資料送到磁碟上去（而不是螢光幕）。一寫入的命令能夠以在PRINT內的DOS 命令來把它取消掉（即使是“空的”DOS 命令（CTRL-D ）也可以）。

有INPUT X$形式的命令也可把寫入命令取消掉，但是只有當把？字元儲入當作主文檔最後一個字元後才可以（在正常情況下INPUT 命令會把？字元顯示在螢光幕上）。在下面指令中INPUT “WHAT'S YOUR NAME?”；X$內字串？字元傳送到磁碟後，WRITE命令會被取消掉。

錯誤訊息也可把WRITE命令取消掉，但是只有在整個錯誤訊息是儲放於你的主文檔最後一欄之後的情況之下才行。

在循序檔內WRITE命令的語法如下：

WRITE f

〔注意：WRITE命令也可用於隨機存取檔，見第8章〕

例子：WRITE LETTER

WRITE RIGHT

在本節開始的幾個例子都是用來說明產生一主文檔所需要的最基本命令。另外有一稍爲一般化，名叫做MAKE TEXT的APPLESOT型程式，它是位於磁碟機內的系統主磁碟上。

MAKE FILE讓你能夠新創一循序檔，這檔可包含到100字串；每個字串最多有239字母。試試它吧一你會喜歡它的。把系統主磁碟放入磁碟機並打入：LOAD MAKE TEXT。

程式之列印如下所示

```
5  REM   MAKE TEXT
10  DIM A$(100):I = 0
20 D$ =  CHR$ (4): REM  CTRL D
30  HOME : TEXT
40  PRINT "YOU GET TO TYPE ONE ST
     RING AT A TIME."
50  PRINT "A STRING MAY HAVE UP T
     O 239 CHARACTERS."
60  PRINT "THIS PROGRAM LETS YOU
     WRITE TEXT FILES.       "
70  PRINT :I = I + 1
80  PRINT "(PRESS THE RETURN KEY
     TO QUIT.)
90  PRINT "TYPE STRING #";I;": ";
100  INPUT "";A$(I)
110  IF A$(I) <  > "" GOTO 60
120  PRINT
130  INPUT "WHAT FILE NAME? ";N$
140  PRINT D$;"OPEN ";N$
150  PRINT D$"DELETE ";N$
160  PRINT D$;"OPEN ";N$
170  PRINT D$;"WRITE ";N$
180  PRINT I - 1
190  FOR J = 1 TO I - 1
200   PRINT A$(J)
210  NEXT J
220  PRINT D$;"CLOSE ";N$
```

一旦程式被載入記憶體後，再把它保存（SAVE）到另一“可寫入”的磁碟中。（此步驟是必須的，因爲這程式就像第 4 章的ANIMAL 程式會產生一檔。）

先檢查一下MAKE TEXT是否仍在你的APPLE 主機中？然後一可寫入磁碟是否已插入磁碟機中？如果是如此的話，打入MONC，I，O這樣一來，你就能看到各種命令如何進出磁碟，再打入RUN 你就會看到下面訊息：

```
YOU GET TO TYPE ONE STRING AT A TIME
A STRING MAY HAVE UP TO 239 CHARACTERS.
THIS PROGRAM LETS YOU WRITE TEXT FILES

(PRESS"THE RETURN KEY TO QUIT.)
TYPE STRING #1:
```

打入任何你喜歡的字串（可以打入100個字串）。注意：此程式使用INPUT命令，所以在字串中不要打入逗點或分號。當你希望停止，回答時就不要打入字串而打入RETURN 鍵，這時你將被問：WHAT FILE NAME？你可爲你的主文檔選一名稱，再按下RETURN 鍵，此時你的字串才被送到磁碟去而你也可從螢光幕上看到它們：首先會出現如下的磁碟命令：

OPEN f

DELETE f

OPEN f

WRITE f

（f代表你所選擇的檔名稱），這些命令後會接著印出一數字，此數字代表所送到主文檔的字串數量（在下一節討論取用此檔的程式將會用到此數字）。再接下來你將看到所打入的字串，最後你將看到此訊息：CLOSE f。執行MAKE TEXT程式後結果如下

```
THIS PROGRAM LETS YOU WRITE TEXT FILES.

YOU GET TO TYPE ONE STRING AT A TIME.

A STRING MAY HAVE UP TO 239 CHARACTERS.


(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING #1: HERE'S STRING 1

(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING #2: AND MY SECOND STRING

(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING #3: ON WE GO

(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING #4: ENOUGH ALREADY!

(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING #5:

WHAT FILE NAME? TEST
OPEN TEST
DELETE TEST
OPEN TEST
WRITE TEST
4
HERE'S STRING 1
AND MY SECOND STRING
ON WE GO
ENOUGH ALREADY!
CLOSE TEST
```

如果你開啓一已經存在的主文檔，然後寫入資料（而非先刪除它再重新開啓它），那麼至少有一部份的檔會比原來輸入的檔要長。除非要蓋掉舊檔的新檔至少和舊檔一樣長，否則結果就會變成新檔後接著舊檔的尾巴。爲了要除去舊檔的所有資料，可先開啓此檔然後刪除它，再開啓一次就可得全新檔（MAKE　TEXT程式中第

14Ø行，15Ø行就是爲去除舊檔資料而給的指令）。如要防止程式寫入資料時要覆蓋到其它檔，可把那些檔鎖制起來。

循序檔的讀出

DOS 命令中的READ可讓你取用一主文檔。一旦READ命令被執行時，以下的INPUT命令（或在APPLESOFT中的GET）就會去參改這特定的檔而不去鍵盤上找資料。此APPLESOFT程式取出名叫SAMPLE的主文檔資料，此檔爲前節開始的那個程式所創。READ命令和WRITE命令一樣，在使用一檔前都要經過開啓的手續。用完後也需要關閉掉。

```
5  REM  RETRIEVE SAMPLE
10 D$ =  CHR$ (4): REM   CHR$(4)
     IS CTRL-D
20  PRINT D$;"OPEN SAMPLE"
30  PRINT D$;"READ SAMPLE"
40  INPUT A$,B$,C$
50  FOR I = 1 TO 10
60  INPUT W
70  NEXT I
80  PRINT D$;"CLOSE SAMPLE"
```

READ命令前必須有個OPEN命令，READ命令後也必須跟著INPUT命令（在APPLESOFT內則爲GET 命令）。OPEN及READ必須參改到同一檔案。如果你執行此程式，並在MON C，I，O的情況下，你將看到：

```
OPEN SAMPLE
READ SAMPLE
?HI HO
??HI HO
??OFF TO THE DISK WE GO
?1
?2
?3
?4
?5
?6
?7
?8
?9
?10
CLOSE SAMPLE
```

此程式是在完全了解SAMPLE檔的情況下而寫的：假設此主文檔包含有3個字串（程式第40行的A$，B$，C$）及10個整數（第60行的W）。當B$及C$被輸入時有兩個問號出現，這是因爲有RETURN鍵分開了這些輸入。READ命令能用PRINT敍述內的DOS命令取消。即使“空的DOS命令（CTRL-D）”也可達此目標。使用PR#或IN#命令也能取消READ命令。

READ的語法和WRITE的相同：

READ f 〔注意：READ命令也可使用於隨機檔中，見第8章〕

例子：READ LETTER

READ CAREFULLY

在APPLESOFT型程式內使用CTRL-C來停止READ命令則會產生REENTER的字串。爲了避免如此，可按RETURN鍵停止執行此程式。

還有一APPLESOFT程式（RETRIEVE TEXT）也是位

於系統主磁碟上，它能取用由MAKE TEXT程式所創的主文檔。用法如下：把系統主磁碟放入磁碟機內後打入：LOAD RETRIEVE TEXT。然後把它儲存到上一次你儲存MAKE TEXT程式的那片磁碟上。（這樣比較方便）。

此程式的到印如下：

```
10 D$ =  CHR$ (4): REM  CTRL D
12  PRINT "THIS PROGRAM RETRIEVES
      TEXT FILES"
14  PRINT "CREATED BY THE 'CREATE
      TEXT' PROGRAM."
16  PRINT "MON C,I,O IS IN EFFECT
      "
18  PRINT
20  INPUT "NAME OF TEXT FILE? ";Z
    $
22  PRINT D$;"MON C,I,O"
24  PRINT
30  PRINT D$;"OPEN ";Z$
40  PRINT D$;"READ ";Z$
50  INPUT I
55  DIM A$(I)
60  FOR J = 1 TO I
70 : INPUT A$(J)
80  NEXT J
90  PRINT D$;"CLOSE ";Z$
100  PRINT D$;"NOMON C,I,O"
```

打入RUN 後你將看到以下訊息：

```
THIS PROGRAM RETRIEVES TEXT FILES
CREATED BY THE 'MAKE TEXT' PROGRAM.
MON C,I,O IS IN EFFECT.

NAME OF TEXT FILE?
```

打入用MAPE TEXT程式所創主文檔的檔名，按下RETURN 鍵後就可。

以下就是把RETRIEVE TEXT程式應用到上一節後面所創

的TEXT檔的結果（即接著鍵入 TEXT ）：

```
THIS PROGRAM RETRIEVES TEXT FILES
CREATED BY THE 'MAKE TEXT' PROGRAM.
MON C,I,O IS IN EFFECT.

NAME OF TEXT FILE? TEST

OPEN TEST
READ TEST
?4
?HERE'S STRING 1
?AND MY SECOND STRING
?ON WE GO
?ENOUGH ALREADY!
CLOSE TEST
NOMON C,I,O
```

更多有關循序檔的資料：

APPEND及POSITION命令

DOS 命令中的APPEND 命令及POSITION命令分別的能使你把主文加到循序主文檔的尾端及取用在主文檔特定一欄所藏的資料。

APPEND 命令允許你加資料到循序主文檔之後。如果想擴張循序主文檔時這條命令就特別有用，在第 4 章ANIMAL程式就是一例。你也許會想起，OPEN命令永遠會把檔定位的指標置爲第 0 拜，也就是檔的第一個字元。而APPEND 命令會幫你先OPEN你所需要的那個檔，然後把檔定位指標指引檔的最後一字元的下一個位置上去。

下面一程式建了一個叫做TESTER 的檔，它包含了兩個字串“ TESTϕ ”及“ TEST1 ”：

```
5  REM   MAKE TESTER
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN TESTER"
30  PRINT D$;"DELETE TESTER"
40  PRINT D$;"OPEN TESTER"
50  PRINT D$;"WRITE TESTER"
60  PRINT "TEST 0"
70  PRINT "TEST 1"
80  PRINT D$;"CLOSE TESTER"
```

下面程式把“TEST2”，“TEST3”，“TEST4”字串附到TESTER 檔之後。

```
5  REM   APPEND TESTER
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"APPEND TESTER"
30  PRINT D$;"WRITE  TESTER"
40  PRINT "TEST 2"
50  PRINT "TEST 3"
60  PRINT "TEST 4"
70  PRINT D$;"CLOSE TESTER"
```

下面程式會把TESTER 程式展現出來：

```
10  REM   RETRIEVE TESTER
20 D$ =  CHR$ (4): REM  CTRL-D
30  PRINT D$;"OPEN TESTER"
40  PRINT D$;"READ TESTER"
50  FOR I = 1 TO 5
60  INPUT A$
70  NEXT I
80  PRINT D$;"CLOSE TESTER"
```

APPEND 命令後必須緊接著WRITE命令（如果接了READ命令只能得到END OF DATA 訊息），APPEND 命令的語法是你非常熟悉的型式：APPEND f〔,Ss〕〔,Dd〕〔,Vv〕

如果檔被鎖制，APPEND 命令仍可把資料附加進去而不會有收到FILE LOCKED 的情況發生。但如果你對鎖制檔使用

WRITE命令，上面情況就會出現。

POSITION命令讓你能夠寫入或讀出任何你想要的主文檔內之欄。POSITION命令的語法如下：POSITION f 〔,Rp〕其中Rp 代表欄在檔中的相對位置。這命令使得檔的定位指標（以現在所指位置爲準）向後移 p 個位置。如果P = ϕ 下面所接的READ及WRITE 命令就從現在所指的欄開始執行。如果P = 1，則READ或WRITE 命令則從下一欄開始執行，如果P = 2 則READ或WRLIE命令就會跳過兩欄（包含現今所指的欄）再執行，以次類推。如果跳到沒有資料存在的欄時，系統會通知你：END OF DATA，程式執行也將停止。

POSITION命令及Pp 參數指定了現今指標所在位置以後第P個欄。POSION 命令必須使用在已經開啓的欄。OPEN命令自動把定位指標放回到檔的第一個位置。因此，如果OPEN命令後緊接著 POSITION命令，那麼Pp 所代表的也就是欄的絕對位置，這在任何狀態下都是對的。

就像其它DOS 命令一樣，POSITION 命令會把READ或WRITE的命令取消掉，因此POSITION命令必須用在READ或WRITE命令之前。

POSITION 命令在執行時，事實上就是一個拜一個拜的去掃描檔的內容，找出第Pp 個RETURN 字元。如果，在此步驟中，遇到了一“空”拜（值爲ϕ），END OF DATA 訊息馬上就會出現。

以下的程式是從TESTER 檔中取出許多欄出來。TESTER檔是由先前MAKE TESTER 及APPEND TESTER程式所創。

```
10  REM   POSITION TESTER
20 D$ =  CHR$ (4): REM  CTRL-D
30  PRINT D$;"OPEN TESTER"
40  PRINT D$;"POSITION TESTER,R2"
50  PRINT D$;"READ TESTER"
60  INPUT A$
70  PRINT D$;"POSITION TESTER,R1"
80  PRINT D$;"READ TESTER"
90  INPUT B$
100  PRINT D$;"OPEN TESTER"
110  PRINT D$;"POSITION TESTER,R3
     "
120  PRINT D$;"READ TESTER"
130  INPUT C$
140  INPUT E$
150  PRINT D$;"CLOSE TESTER"
```

如在MON C，I，O情況下執行此程式，你將看到：

```
OPEN TESTER
POSITION TESTER,R2
READ TESTER
?TEST 2
POSITION TESTER,R1
READ TESTER
?TEST 4
OPEN TESTER
POSITION TESTER,R3
READ TESTER
?TEST 3
?TEST 4
CLOSE TESTER
```

對以上C$，E$，的結果感到驚奇嗎？注意OPEN後檔定位指標就指向ϕ，還有，每個INPUT命令只能使一欄讀入APPLE主機內，讀完後定位指標自動指到下一欄之起始處。

拜定位後處理資料的方法

注意：本節非爲初學者而寫，用以上的資料就足夠良好的使用循序檔而不必用到下面所討論到的資料。

DOS 命令中的READ及WRITE能夠與一拜參數（Byte parameter）合用來從一主文檔的任何地方寫入或讀出資料（如果你知道資料在那裏的話）。首先你必須知道從檔中那一拜開始，爲此，你又需先知道資料是如何被儲放到此檔中。你必須先數出所有RETURN，逗號，空白及其它字元數量然後才知道要從何方開始。執行WRITE更困難，因爲你必須要知道那裏停止才行。

B參數（即拜參數）指的是在檔中的絕對位置，但如果有個R（relative相對的），則B參數成爲某一特定欄的眞實位置。

WRITE THISMONTH，B27"命令把檔指標指到THISMONTH檔的第28 拜位置（注意：第一拜的編號爲 ϕ）。緊接下來的PRINT命令會把資料傳送到磁碟，並從第 28 拜開始以此新資料取代掉原有的資料。

這種以新資料取代舊資料並不只是修改定位指標所指的欄之資料而已。如果你所印的新資料比原有資料要短，那麼此欄將變成兩個欄：其中之一就是剛印入的新資料那一欄，另外則是舊資料尾巴未被新資料蓋到的亦自成一欄。又如果新資料比原有資料要長的話，那麼就會侵入下一欄。如此一來，指標所指的那一欄比原先的要長，而其下一欄則較短。

也有可能發生寫入的那個拜剛好是在檔最後一個拜之後。企圖讀出這因超長而寫不入的拜將會使END OF DATA 訊息出現，你的程式也會停止。參看READ 指令伴同B參數時，取用循序檔內不相鄰欄那一部份的討論。

此命令的語法如下：

WRITE f 〔,Bb〕

其中B參數指定了檔中第B + 1個拜爲下個PRINT命令資料

所要取代原資料的地方。b的自然值（default value）爲ϕ，代表檔的第一個拜。第b拜代表此檔在記憶裝置中的眞實或絕對的位置。B參數可以指定指標以前或以後的拜當作寫入的起點。〔注意，此命令也可用在隨機存取檔中，見第8章〕。

相同的，以下的命令

READ LASTMONTH , B32

把指標指到LASTMONTH檔第33 拜上去（第一拜的編號爲ϕ）。接下來的INPUT命令會將第33 拜開始那一欄的所有字元讀入APPLE主機中。如果第33 拜不是那一欄的起首，那麼只有第33 拜及以後部份的欄資料被輸入。

此命令之語法爲

READ f 〔,Bb〕

其中B參數指定了下一個INPUT或GET 命令所要讀入的字元，b的自然值爲ϕ，代表檔的第一個拜。b是在檔在記憶裝置中眞實或絕對的位置。B參數也可指到指標以前或以後的資料去。〔注意：此命令也可用在隨機存取檔，見第8章〕。

下面的程式把TESTER 檔定位指標指到第14 拜的位置（TESTER 檔爲以前MAKE TESTER 程式所創）。然後它再寫入“APPLE COMPUTER ”字串。注意下列順序：OPEN →WRITE及PRINT →CLOSE。

```
5  REM   BYTE WRITER
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN TESTER"
30  PRINT D$;"WRITE TESTER,B14"
40  PRINT "APPLE COMPUTER"
50  PRINT D$;"CLOSE TESTER"
```

在MON C，I，O之情況下，執行RETRIEVE TESTER

程式，看看此上程式如何修改TESTER 。你將看到包括APPLE COMPUTER 之欄完全取代了包含TEST3 及TEST4 資料的兩欄。及包含TEST5 那欄的第一個字元。執行完後總共只剩四欄，END OF DATA 訊息在第5 個INPUT 命令後出現。

下面的程式把TESTER 檔指標定在第18 拜上（只是對前一程式稍作修改）。這程式將從此拜讀入資料直到遇到RETURN 字元。順序如下：OPEN →READ →INPUT 或GET → CLOSE 。

```
5  REM   BYTE READER
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN TESTER"
30  PRINT D$;"READ TESTER,B18"
40  INPUT A$
50  PRINT D$;"CLOSE TESTER"
```

在執行此程式前，可試一試去預測結果到底如何。

第七章

自動化的 APPLE

利用一個主文檔控制APPLE: EXEC命令
新建一個EXEC 檔

在主文檔中截取程式
將機器語言常式轉換爲BASIC語言
執行(EXEC)的過程

想要深入了解本章的內容，最好先參閱第六章所介紹的循序主文檔。

利用一個主文檔控制APPLE：EXEC

DOS 的命令EXEC跟RUN 很像，只差在 EXEC 是用在其中含有命令或程式（包括BASIC指令）的主文檔，而使這些命令或程式指令就像從鍵盤輸入一樣。

欲求瞭解EXEC命令的功能，可先對系統的主磁碟（System Master）執行LOAD EXEC DEMO，將EXEC DEMO檔載入記憶器中，並保存（SAVE）在一個無寫入保護的磁碟上。因爲此一程式會寫出（WRITE）一個主文檔，所以要將此無寫入保護的磁碟放在推動器內。

然後執行（RUN）此程式。將出現下列訊息：

```
<<  EXEC DEMO  >>

THIS PROGRAM CREATES A SEQUENTIAL TEXT
FILE NAMED "DO'ER" CONTAINING SEVERAL
STRINGS, EACH A LEGAL APPLE II COMMAND.
WHEN YOU TYPE
EXEC DO'ER
THEN THE COMMANDS IN FILE DO'ER TAKE
CONTROL OF YOUR COMPUTER  EACH COMMAND
WILL BE EXECUTED JUST AS IF IT HAD BEEN
TYRED AT THE KEYBOARD.  THE DOS MANUAL
DESCRIBES THE PROGRAM IN MORE DETAIL.
```

≪EXEC DEMO≫
此程式新建一個循序主文檔，即DO'ER，其中包含一些字串，每一字串都是合法的APPLEⅡ命令。當你鍵入EXEC DO'ER時，DO'ER檔中的命令即取得系統的控制權。其中每個命令就像從鍵盤輸入一樣。DOS 手冊

對於此程式有進一步的說明。

```
<< HAPPY EXECUTING >>
PRESS THE SPACE BAR TO MAKE THIS
PROGRAM CREATE THE FILE DO'ER.
IF YOU WISH TO STOP THIS PROGRAM NOW,
YOU MAY PRESS THE ESC KEY.
```

≪愉快的執行≫

按下SPACE鍵，即可開始令此程式建立DO′ER檔。若要停止此程式，則按下ESC 鍵即可。

按下SPACE 橫鍵之後，稍停一會兒，即可看到磁碟機的IN USE（使用中）燈會亮，表示此程式正在將DO′ER檔寫入磁碟。然後鍵入

EXEC DO′ER

並按 RETURN鍵。則系統會根據DO′ER檔中定好的命令開始動作。

DO′ER的動作情形大致是這樣的：

首先發出MON C，I，O命令，以便能讓你看到動作情形。

其次，將一個僅有三行的程式寫入，並以NEW PROGRAM!! 的名字存入磁碟。然後將此程式列印（LIST）出來。

接著執行一個FOR-NEXT環路以延遲一段時間，讓你在下一動作開始之前看清顯示幕上的訊息。

接著，DO′ER利用 INT 命令進入Integer BASIC，載入（LOAD）COLOR DEMOS 程式，並列印出來。

之後，DO′ER利用CALL-151 進入監督程式，並在利用FP 命令進入Applesoft 之前執行一些機器語言的指令。

在Applesoft 中先執行MON C，I，O命令，然後再執行、修改並印出NEW PROGRAM!!（再執行一個環路動作，以便

看清顯示幕上的訊息），並取名爲EVEN MORE RECENT PROGRAM!! 予以保存。

最後，將程式NEW PROGRAM!!予以刪除（DELETE）後，顯示檔案目錄（CATALOG）（其中包括有新加入的EVEN MORE RECENT PROGRAM!! 程式）。

這些動作均自動完成，甚至不須按下任何鍵（除非檔案目錄中已超過18 個進入點，則須按SPACE鍵檢查目錄中進入點的使用狀況）。

新建一個EXEC檔

下面要用一個例子來逐步說明如何建立一個名叫DOIT 的EXEC檔，其中的命令內容爲：

```
LIST 20, 50
RUN AWAY
CATALOG
```

首先建立並保存一個叫做AWAY 的Applesoft 程式：

```
5  REM  AWAY
10  PRINT "AWAY AWAY WITH RUM BY
     GUM"
```

其次，寫入並保存下面的程式MAKE EXEC，此程式將建立一個主文檔DOIT。在以後做EXEC DOIT時，MAKE EXEC程式所產生到DOIT主文檔（經由PRINT指令）中的命令會指示系統執行AWAY 程式。須注意在DOIT檔案中各命令之前均不加CTRL-D 。

```
5  REM   MAKE EXEC
10 D$ =  CHR$ (4): REM CHR$(4) IS
      CTRL-D
20  PRINT D$;"OPEN DOIT"
30  PRINT D$;"WRITE DOIT"
40  PRINT "LIST 20,50"
50  PRINT "RUN AWAY"
60  PRINT "CATALOG"
70  PRINT D$;"CLOSE DOIT"
```

在將MAKE EXEC及AWAY 經由SAVE命令存入磁碟之後，鍵入RUN MAKE EXEC
以建立一個循序主文檔DOIT。

鍵入下面的命令

EXEC DOIT

將使得DOIT檔中的命令一個接一個的執行，恰如從鍵盤輸入一樣。須再注意此處的各命令在從MAKE EXEC程式產生時，其命令之前均不加CTRL-D 。先將目前記憶體中的程式(可能爲MAKE EXEC程式）前20至50行予以列印出來。然後，執行 AWAY 程式，最後，將磁碟上的目錄顯示出來。

在主文檔中截取程式

再介紹一個極有用的例子，說明如何使用EXEC命令，可將列印出來的程式當作主文檔。此一程式可用以：

* 將Integer BASIC程式轉換爲Applesoft 程式。
* 將部分程式予以重新編號，並即執行（EXEC），產生的結果存入另一程式中。
* 從磁碟上的副程式檔中選取合用的副程式插入任何程式。
* 將一程式附加到另一程式之後。

* 修補任一程式中已變得不可讀的部分（可從一主文檔中截取良好的部分，予以重新起動，然後經由EXEC而將此部分放回記憶體中）。

在下面的CAPTURE 程式中第六行的LIST命令之後的行號數。2270 及 5130 要用所欲截取而存在記憶體中的程式的行號數來取代，含有這些列印的循序主文檔名稱爲LISTING。

```
1  REM  CAPTURE
2 D$ =  CHR$ (4): REM  CTRL-D
3  PRINT D$; "OPEN LISTING"
4  PRINT D$; "WRITE LISTING"
5  POKE 33,30
6  LIST 2270,5130
7  PRINT D$; "CLOSE LISTING"
8  TEXT : END
```

上面程式的行號數很靠近，以便於將其加到目前在記憶體中的程式裏，或程式中任何有八個間隔行號數的中間。也可以很簡單地把CAPTURE 全部放在程式的最後一行之後。

CAPTURE 建立一個帶有行號數的命令主文檔。在執行（EXEC）此主文檔時，帶有行號數的命令並不會執行。正如由鍵盤輸入一樣，這些命令行是當作程式一般存在記憶體中的。而在當其被截取到一個主文檔中，則可加以修改並經執行（EXEC）後存回記憶體中。與LOAD或 RUN 不同的是，EXEC不會刪除已在記憶體中的程式。利用 CAPTURE ，可將某一語言的程式截取到一個主文檔中，加以執行（EXEC），變爲另一種語言送回（因爲程式若不稍加修改，就可能無法執行一譬如說Integer BASIC 與Applesoft的語法就有些不同）。利用EXEC也可把新行加到目前記憶體中的程式上去。事實上，你可把CAPTURE 程式當作是一

個主文檔LIST　SAVER 予以保存，即可執行EXEC LIST SAVER任意次，以將CAPTURE 程式加到一個正在記憶體中的程式內。

將機器語言常式轉換爲BASIC語言

下面再介紹另一個有用的程式，可把機器語言寫成的常式直接塞進（利用POKE指令）記憶體中，而變成BASIC程式的一部份。此一程式部分可以當作Applesoft 或Integer BASIC程式的一部分，在每次當BASIC程式執行時，即將機器語言常式放進記憶體中。

```
5  REM  CODE-POKES WRITER
10 D$ = "": REM   CTRL-D
15  PRINT D$;"OPEN CODE-POKES"
20  PRINT D$;"DELETE CODE-POKES"
25  PRINT D$;"OPEN CODE-POKES"
30  PRINT D$;"WRITE CODE-POKES"
40 LINENUMBER = 7000
50  FOR PLACE = 768 TO 783
60 COUNTER = COUNTER + 1
70  IF COUNTER = 10 THEN COUNTER =
    1
80  IF COUNTER <  > 1 THEN 120
90  PRINT
100  PRINT LINENUMBER;
110 LINENUMBER = LINENUMBER + 1
120  PRINT " POKE ";PLACE;",";  PEEK
    (PLACE);" :";
130  NEXT PLACE
135  PRINT
140  PRINT D$;"CLOSE CODE-POKES"
150  END
```

在使用此程式時，第4ϕ 行中的數目要用BASIC程式中POKE程式開始部分的行號數取代。第5ϕ 行中的FOR 環路中

應包含欲轉換進來的機器語言常式開始和結束位址（十進位表示）。

當程式鍵入之後予以執行（RUN），會產生主文檔CODE-POKES。再利用下面的命令

```
EXEC CODE-POKES
```

將使機器語言的“POKE”程式部分放進另一程式中經事先指定好的行號數之後。CODE-POKES WRITER程式可以配合Applesoft或Integer BASIC使用。

MAXFILES 和INTEGER BASIC程式

若想要從Integer BASIC內部來增加MAXFILES而不致抹除掉程式，則須使用EXEC檔。先用前面的方法建好一個EXEC檔，暫時取名爲FILE.EX。

FILE.EX包含下列的命令，可允許五個檔同時在系統上：

```
MAXFILES 5
LOAD PROGRAM
DEL 1φ,2φ
RUN
```

此程式的前幾行如下所示，其中CTRL-D是由同時按下CTRL及D鍵而產生。

```
1φ PRINT "CTRL-D EXEC FILE.EX"
2φ END
3φ REM MAIN PROGRAM
```

執行(EXEC)的過程

EXEC命令的語法一般如下：

EXEC　f

其中 f 是包含有BASIC指令或程式的循序主文檔名稱。前面幾節中對此已有舉例說明。在此EXEC命令中讀入 f 檔案的第一段，就跟從鍵盤輸入一樣。當碰到“RETURN”字元時，Apple會將此段內容當作BASIC的命令或程式予以執行。BASIC的型式（Integer 或Applesoft）不會受EXEC 改變，除非在檔案中包含有FP或INT 命令。當第一段執行結束時，繼續讀入 f 中的第二段予以執行，依此繼續，直到最後一段。

EXEC命令不會被CTRL-C 鍵所停止。

在任何時刻，僅有一個EXEC可以被開檔（OPEN）。若正有一個EXEC檔在執行，而其中又恰有一個EXEC命令要執行，則第一個EXEC檔立刻關閉（CLOSE）。然後，變成第二個EXEC命令在執行。

若一個正在執行的檔中有執行（RUN）任何程式的命令，則EXEC會耐心的等待此程式執行結束。然後，再繼續執行下一個EXEC檔案命令。

然而，若一個EXEC 檔正好是開始的，而正在執行一個程式則此程式中的任何INPUT指令會將此正在執行中的檔案下一欄當作是它的回答，而不理鍵盤的輸入。更糟的是，若此回答是一個立即執行的DOS 命令，此命令將會在程式繼續下去之前即予執行，而產生出乎意料之外的結果。

若按下CTRL-C鍵使一個正在執行的Applesoft 程式中止而此時正有一個EXEC檔是開啓的，則此EXEC檔剩下的部分將

不會執行。

若一EXEC檔中有任何無效的命令或程式，則會出現以下的訊息：

SYNTAX ERROR（語法錯誤）

而再讀取下一欄繼續。因此，對任何主文檔，不管是否包含BASIC程式指令，皆可執行（當然，首先須確定程式保存在記憶體中）。在有MON C，I，O的情形下，這個方式可提供一個粗略但很方便的工具以供迅速的檢視一個主文檔的內容。

EXEC命令也可使用相對欄位的參數，而與POSITION 命令中所使用的略有不同。其命令的語法爲

EXEC f 〔,Rp〕

其中Rp 指定從檔案 f的第P 欄開始執行。因爲EXEC通常設定其檔案位置指標在檔案的第一個字元，所以Rp 通常指的是相對於檔案開始位置的第 p 欄。p 通常與檔案的實際或絕對欄位相對應。Rφ指示EXEC由檔中的第一欄開始，R1 指示由第二欄開始……依此類推。

須注意此處R參數與POSIPION中的用法不同，在後者中的R3僅表示相對欄位，而有可能在程式中不同的場合指向不同的實際欄位。

EXEC MYFILE，Rf

當R參數所指定的是在檔案結束之後的第二個欄位，則會產生END OF DATA的訊息。（若所指定的是在檔案結束之後的第一個欄位，則無訊息。）

第　八　章

隨機存取檔的用法

隨機存取檔：如何工作

一個特定的資料錄

多重資料錄

一個隨機存取檔的範例：RANDOM程式

對隨機存取檔的寫入與讀取

欲對本章深入瞭解，最好請先參閱第六章的循序檔。

隨機存取檔：如何工作

隨機存取主文檔就像是由許多大小相同的蜂巢室集合成蜂巢一樣－其中有些是滿的，有些則是空的。每一個蜂巢室就像一個資料錄，當新建一個隨機存取檔時，必須先指定其中所含資料錄的標準尺寸大小。

隨機存取檔中的資料錄都是固定長度，與循序檔中可爲任意長度不同。當第一次要將一特定的資料錄寫入檔中時，會先在磁碟上安排足夠的空間，以供容放一個完整的標準長度資料錄，而不管此資料錄實際上是否填滿。因此，隨機存取檔不一定是最有效率的使用空間。但是，因爲這種檔案的建立方式相當有規則，所以用在存取或修改時，相當快速簡易。

隨機存取檔適合用在須快速存取其任何部分的檔案，或其資料經常變動的檔案。例如，對郵寄清單就非常適合用隨機存取檔來處理。

隨機存取檔的建立和存取方式與循序檔頗爲類似。主要的差異在某些命令有特別的參數，例如OPEN中須要一個L（Length of record），READ 和WRITE中都有R（Record 號數）參數。在詳細說明建立與存取隨機存取檔的方法以及新參數如何作用之前，先介紹幾個例子。有關較技術性的使用方法請參閱附錄C。

一個特定的資料錄

如何從隨機存取檔中存取一個特定的資料錄呢？下面幾個

Applesoft 程式說明如何利用DOS 達到此目的。第一個程式請求輸入名稱（N$），電話號碼（P$）和郵遞區號（Z$）然後將這些資料送入檔案MAILER 的資料錄1：

```
10  REM     MAKE MAILER
20 D$ =  CHR$ (4): REM  CTRL-D
30  INPUT "NAME:       ";N$
40  INPUT "PHONE:      ";P$
50  INPUT "ZIP CODE:  ";Z$
60  PRINT D$;"OPEN MAILER,L200"
70  PRINT D$;"WRITE MAILER,R1"
80  PRINT N$: PRINT P$: PRINT Z$
90  PRINT D$;"CLOSE MAILER"
```

第2φ 行把CTRL-D 放在變數D$

第3φ 到第5φ 行請求輸入欲儲存的資料。

第6φ 行開啓一個叫MAILER的檔，其中每個資料錄的長度爲2φφ個拜。

第7φ 行準備好將資料記錄到資料錄1中。

第8φ 行將N$，P$，Z$送到磁碟——在第7φ 行中已指定爲資料錄1，三欄之間用RETURN 隔開。

第9φ 行關閉該檔。

在有MON C，I，O命令的情況下，執行此程式會顯示出：

```
NAME:       AMY DOAKS
PHONE:      (425) 555-1010
ZIP CODE:  95014
OPEN MAILER,L200
WRITE MAILER,R1
AMY DOAKS
(425) 555-1010
95014
CLOSE MAILER
```

檔案MAILER 的資料錄1可經由下面的程式存取：

```
10  REM  RETRIEVE MAILER:A
20 D$ =  CHR$ (4): REM  CTRL-D
30  PRINT D$;"OPEN MAILER,L200"
40  PRINT D$;"READ MAILER,R1"
50  INPUT N1$,P1$,Z1$
70  PRINT D$;"CLOSE MAILER"
```

在有MON C，I，O命令的情況下，執行的結果如下。其中出現的雙疑問號表示其INPUT不止有一個回應。

```
OPEN MAILER,L200
READ MAILER,R1
?AMY DOAKS
??(425) 555-1010
??95014
CLOSE MAILER
```

下面爲另一個稍有不同的程式來存取資料錄1：

```
10  REM   RETRIEVE MAILER:B
20 D$ = "": REM  QUOTES CONTAIN
     CTRL-D
30  PRINT D$;"OPEN MAILER,L200"
40  PRINT D$;"READ MAILER,R1"
50  INPUT N1$
60  INPUT P1$
70  INPUT Z1$
80  PRINT D$;"CLOSE MAILER"
90  END
```

多重資料錄

前面的程式建立了一個隨機存取檔MAILER，並寫入了一筆資料錄，其中保存了三項以RETURN字元分開的資料。下面的程式說明如何將資料寫入多個資料錄，在此例中是寫入隨機存取檔RA-FILE 的第12 到第15 個資料錄。

```
5  REM  MAKE RA-FILE
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN RA-FILE"
30  PRINT D$;"DELETE RA-FILE"
40  PRINT D$;"OPEN RA-FILE, L30"
50  FOR I = 12 TO 15
60  PRINT D$;"WRITE RA-FILE, R";I
70  PRINT "NAME ADDRESS ";I
80  NEXT I
90  PRINT D$;"WRITE RA-FILE, R13"
100  PRINT "DOS"
110  PRINT D$;"CLOSE RA-FILE"
```

第1φ 行設定D$ 爲CTRL-D 。

第2φ 和第3φ 行確使RA-FILE 爲新檔。

第4φ 行開啓RA-FILE檔，其中每一筆資料錄的長度都是3φ個拜

第5φ 到第8φ 行構成環路，將字串"NAME ADDRESS"及所跟的資料錄號數（ 12 到15 ）寫出。須注意在PRINT之前，須先指定好WRITE中的新資料錄。

第9φ 和第1φφ行將資料錄13 的資料改變成第1φφ行PRINT指令中所給的文字。

第11φ行關閉隨機存取檔RA-FILE。

在有MON C，I，O命令的情況下執行此程式，則可看到以下結果：

```
OPEN RA-FILE
DELETE RA-FILE
OPEN RA-FILE, L30
WRITE RA-FILE, R12
NAME ADDRESS 12
WRITE RA-FILE, R13
NAME ADDRESS 13
WRITE RA-FILE, R14
NAME ADDRESS 14
```

```
WRITE RA-FILE, R15
NAME ADDRESS 15
WRITE RR-FILE, R13
DOS
CLOSE RA-FILE
```

以相同方式可從某一特定的資料錄或是一主文檔的資料錄中讀取資料。下面的程式即爲從RA-FILE 檔中取出資料錄12到15，並在第60行時，嘗試找出前三個字是“DOS”的資料錄。

```
5  REM  RETRIEVE RA-FILE
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN RA-FILE, L30"
30  FOR J = 12 TO 15
40  PRINT D$;"READ RA-FILE, R";J
50  INPUT A$
60  IF  LEFT$ (A$,3) = "DOS" THEN
      PRINT "RECORD ";J;" WAS CHA

70  NEXT J
80  PRINT D$;"CLOSE RA-FILE"
```

第10行設定D$爲CTRL-D。

第20行開啓主文檔RA-FILE，其中的資料錄皆爲30個拜長（此爲前面在建檔時所指定的長度）。

第30行到第70行讀取RA-FILE 的第12 到第15 筆資料錄。在每次INPUT要從資料錄讀取字元之前，須先指定好READ中的資料錄。在第50行，每一筆從磁碟進來的資料錄都看作是ASCII字串，其結尾是RETURN 。第60行檢查從資料錄r輸入的字串A$ 的最左3位是否爲“DOS”。若是，則印出“RECORD r WAS CHANGED.”並繼續尋找。

第80行關閉該檔。

在有MON C，I，O的命令條件下，執行此程式將產生以下結果：

```
OPEN RA-FILE, L30
READ RA-FILE, R12
?NAME ADDRESS 12
READ RA-FILE, R13
?DOS
RECORD 13 WAS CHANGED
READ RA-FILE, R14
?NAME ADDRESS 14
READ RA-FILE, R15
?NAME ADDRESS 15
CLOSE RA-FILE
```

在讀取檔案時，只曾檢查前面已寫入的資料錄。因此若查問RA-FILE 檔中第 8 筆資料錄，將會收到END OF DATA （資料結束）的訊息，此乃因爲檔中並未寫入該筆資料錄。

同樣的，若想要從目前已有的資料錄中讀取超出其欄位的資料，就會有以下訊息出現：第 12 到第 15 筆資料錄皆僅含一欄資料。

一個隨機存取檔的範例：RANDOM程式

最後來提一下在System Master磁碟中一個叫做RANDOM的程式，這個程式利用一個隨機存取主文檔對一個小型庫存控制系統的運用作一個示範的實例。此程式最多僅可處理 9 個零件。如此可使程式極簡化。事實上，Apple處理上千個零件也仍然是可以的。

首先，這個程式將其本身及控制庫存的隨機主文檔APPLE PROMS複製到記憶器中，然後自動開始執行。你可列出其中一項或全部的存量，也可用一次一個或一次全部的方式以改變項目。下面說明其動作情形。

1. 利用System Master 磁碟

   RUN RANDOM

就出現下列訊息：

```
THIS DEMONSTRATION WILL NOT EXECUTE ON
A WRITE-PROTECTED DISKETTE SUCH AS
YOUR DOS SYSTEM MASTER
FOR YOUR CONVENIENCE, PROVISIONS HAVE
BEEN MADE TO COPY THIS PROGRAM AND IT'S
DATA TO ANOTHER DISKETTE.

DO YOU WISH TO DO THIS NOW? (Y OR N) Y
```

若不要（“ NO ”）則鍵入 N，即可回到 Applesoft 。

2. 若要，則鍵入“ Y ”。此時會出現

```
NOW READING DATA.
```

及

```
INSERT AN INITIALIZED DISKETTE, THEN
PRESS THE RETURN KEY TO BEGIN TRANSFER
```

3. 用一片沒有寫入保護的磁碟來換下 System Master，然後按下 RETURN 鍵，即會出現下列訊息，並開始執行。

```
WHEN THE PROGRAM AND DATA HAVE BEEN
FULLY TRANSFERRED, THE PROGRAM WILL
BEGIN RUNNING.
```

4. 執行過程中，先會顯示出：

```
            APPLE PROMS
COMMAND         NUMBER
-------         ------
LIST              1
CHANGE            2
EXIT              3
CHOOSE NUMBER (1 - 3) 1
```

按“ 1 ”，即出現

```
PART NUMBER 1-9 (0=ALL) 0
```

5. 按 ϕ ，可得到此存量系統中所有零件的清單

```
PART#  NAME             SIZE    IN STOCK
-----  ----             ----    --------
  1    PARALLEL PRINT   256     500
  2    COMMUNICATIONS   256     1250
  3    (NOT AVAILABLE)  256     0
  4    (NOT AVAILABLE)  256     0
  5    DISK BOOT        256     432
  6    STATE MACHINE    256     460
  7    SERIAL PRINTER1  256     878
  8    SERIAL PRINTER2  512     741
  9    CENTRONICS       256     1290

PRESS THE RETURN KEY TO CONTINUE
```

按RETURN鍵，可再回到功能選擇表去。

6. 試試程式中的其他功能選擇表。選1可以根據零件編號一次列出一個或全部零件。

選2可改變任一或所有的零件名稱及描敍。譬如，假設零件3應爲COSMIC GLUE，基準量爲56，存量爲1234其修改方法爲：

選擇功能2，CHANGE

選擇零件編號爲3

則舊的零件名稱會顯示出來，同時游標（Cursor）指在開頭處，此時可打入新的名稱，按下RETURN，則游標右移指向基準量。同理可將新的零件基準量和存量打入。

選3可使此程式終止。

對隨機存取主文檔的寫入與讀取

在隨機存取檔中的CLOSE命令，其作用與循序檔是完全一樣的（見第六章）。但對OPEN命令，則須有另一參數L，其語法爲

OPEN f，Lj〔，Ss〕〔，Dd〕〔，Vv〕

此處L代表資料錄長度，j表示每一資料錄所佔的拜數，若省略不寫，則j被當作是自然值1。j的範圍在1至32767之間。

若在讀取之前開啓檔案時，指定的長度參數與當初寫入時所指定的不同，則DOS會利用新的長度來計算資料錄在檔中的位置。因此，使用者必須將檔案的結構與內容記下備忘（甚至可存在檔中的第ϕ筆資料錄中）。將長度參數包含在檔名中也不失爲一好辦法，如：

RANDFILES：L2ϕ

STOCKLISTS-L1ϕϕ

DIRECTORIES（L5ϕ）等等。

從隨機存取檔中是無法看出資料錄的長度的，所以必須將此長度值記下以備查。

資料錄決不可比L參數所指定的長度還長，否則會蓋寫掉其他的資料或產生怪異的結果。

在WRITE和READ中都有R參數，可用在建立或讀取隨機存取檔中特定的資料錄。

WRITE f〔，Rr〕

READ f〔，Rr〕

例如：WRITE LEGIBLY，R3，

READ FAST，R13

此處Rr（Record）參數是用在建立（當WRITE時）或讀取（當READ時）檔中第r筆資料錄。r省略時的自然值爲ϕ，所指的是第一筆資料錄。

在Applesoft 中用CTRL-C 來停止READ命令會造成一連串的REENTER （重新輸入），正確的應該是按RESET才對。

有時候，隨機主文檔中的每一筆資料錄可分別看作是短短的循序檔。WRITE和READ命令中除了R參數之外尚有B（Byte）參數。B參數指定資料錄在以後的PRINT（WRITE之後），INPUT或GET （READ之後）等指令中的開始拜位。

WRITE f〔,Rr〕〔,Bb〕

READ f〔,Rr〕〔,Bb〕

若指定B參數，則在WRITE（或READ）時會從所指定資料錄的第b拜開始。b省略時的自然值爲ϕ，當作爲第一個拜。B參數亦可指定目前檔案指標位置之前或之後的資料錄。所以在使用B參數時，須對檔中每一筆資料錄的內容完全瞭解。

在READ或WRITE將檔案位置指標指向某一資料錄後，可再用POSITION命令將指標前移（只能前移）到該資料錄中相對的欄位置。使用POSITION會將原先的WRITE或READ模式取消（並不改變檔案位置指標），所以若要再回到原先模式，須再用WRITE或READ命令（不加參數）。

有關如何將一般資料或隨機存取檔儲存在磁碟上，請參閱附錄C。

第　九　章

機器語言檔的用法

機器語言檔

BSAVE

BLOAD

BRUN

RWTS副常式

機器語言檔

DOS 可將記憶體中的資料存入磁碟或從磁碟上讀取出來。SAVE，LOAD，RUN 等DOS 處理記憶體的命令也使用在BASIC的程式中。在這一章裏將討論BSAVE，BLOAD，BRUN等DOS 命令，這些命令執行類似的功能，只是所處理的是記憶體中二進位及十六進位原始形式的資料。

此處幾個命令之前的B表示是處理Binary（二進位）檔，在目錄（CATALOG）中二進位檔的檔名之前也都加B以資識別。二進位檔實際上就是記憶體中特定範圍內的資料，完完全全，一比次一比次的抄錄下來的。這些特定範圍內的資料可能是機器語言程式，二進位資料，或高解像度圖形顯示的比次對映圖像。

BSAVE

BSAVE命令新建一檔f，將一段記憶體的內容存入。語法爲

BSAVE f，Aa，Lj〔，Ss〕〔，Dd〕〔，Vv〕

其中S，D，V等參數分別代表槽路推動器、册號。注意A和L參數不是任選的。

A參數指定要存入磁碟的記憶體開始位址（以十或十六進位表示）。表示十六進位時，位址之前須加$符號。若a小於ø或大於65535，則會顯示

SYNTAX ERROR（語法錯誤）訊息。

因此，對等的負值位址不可用在此命令中。又所指定的位址雖然在ø到 65535之間，但所指定的開始位址實際上並未安裝有記憶

晶片（譬如在48K系統時最大的記憶位址爲 49151或 $BFFF），則並不會有錯誤訊息產生。但是，在此情況下的A參數是無效的。

參數L指定欲儲存的記憶體總拜數。若其值小於 ϕ 或大於 65535，則顯示錯誤訊息

SYNTAX ERROR

若L值爲 ϕ 或在 32767到 65535的範圍內，則顯示訊息

RANGE ERROR （範圍錯誤）。

因爲 32767是可以存入磁碟上單一欄內的最大拜數。所以若想存入比 32767還多的拜數，則須使用二個BSAVE 命令。對於從1到32767之間的L值，即使其中有的位址實際上並未裝有記憶晶片，都是不會有錯誤訊息產生的。當然，指定值若超出實際上的最大記憶位址（譬如 48K系統爲 49151或 $BFFF），都是無效的。

下面這些例子都是新建一個包含有高解像度第二圖像記憶區影像的檔PICTURE。都有同樣的作用，只是其開始位址和長度參數用不同的形式表示。

```
BSAVE PICTURE, A$4000, L$2000
BSAVE PICTURE, A16384, L8192
BSAVE PICTURE, A16384, L$2000
BSAVE PICTURE, A$4000 L8192
```

BLOAD

BLOAD命令將二進位檔的內容送回記憶體來。通常這個命令並不會抹蓋掉已在記憶體中的BASIC程式，除非所指定的記憶體範圍剛好包含了該程式。

此命令的語法爲

BLOAD f〔,Aa〕〔,Ss〕〔,Dd〕〔,Vv〕

其中S，D，V等參數和前面的一樣。若有指定A參數，則二進位檔的內容會載入到從a位址開始的記憶體內，而取代了原有的內容。若未指定A參數，則該檔的內容會載入到當初BSAVE時所指定的原記憶位址內。（請看BSAVE中對A參數的說明。）

假設二進位檔PICTURE 包含一高解像度的圖像。下面這些例子都是將此圖像放進高解像度第一圖像記憶區的記憶體中：

BLOAD PICTURE，A8192

BLOAD PICTURE，A$2000

此處例子執行的結果會將RAM 中的Applesoft 破壞掉。

注意：若機器語言程式被載入的記憶體位址與其當初所指定下的記憶體位址不同的話，則無法執行。

BRUN

BRUN命令的語法與BLOAD一樣：

BRUN f〔,Aa〕〔,Ss〕〔,Dd〕〔,Vv〕

此處的二進位檔應爲機器語言程式。

BRUN首先執行BLOAD。若有指定A參數，則檔的內容會放入從a開始的記憶體內。若未指定A參數，則該檔的內容會載入到當初BSAVE時所指定的原記憶位址內。（請看BSAVE中對A參數的說明。）

執行了BLOAD之後，BRUN接著執行機器語言跳越（JMP）到a的指令。若此檔爲機器語言程式，即開始執行。

RWTS副常式

通常，使用者是透過DOS 來存取DISKⅡ（二號磁碟機）的。但是，利用機器語言可透過另一途徑來存取DISKⅡ。若讀者存對機器語言不熟悉，則可跳過此節不看。

利用DOS 中的副常式RWTS（Read or Write a Track and Sector），即可直接以機器語言存取DISKⅡ。

每一片經由DISKⅡ磁碟機所起始設定（Initialized）的磁碟可分爲35 條軌道，由φ編到34。這些軌道可想像成唱片上的溝紋，只是之間並不連續。基本上，軌道是由許多同心圓組成，中心的大洞爲其共同圓心。軌道φ在最外層，而軌道34 在最內層。磁碟機的讀寫頭就像唱機的唱針，只是前者是磁性的。讀寫頭會移動到不同的軌道上進行資料的讀寫。

每一條軌道包含16 個段落。各段落都經事先劃分爲256個拜組塊當作使用上的基本單位，以免每次都以整個軌道的4φ96 拜爲單位來使用。各段落分別從φ到15 相連編號。當磁碟旋轉時，每一個段落都曾從讀寫頭下通過，此時讀寫頭即可寫入或讀出資料。每一個段落都包含兩部分：即位址欄和資料欄。位址欄中包含有軌道、段落及册號，資料欄中包含有256拜經過重組編排的資料。

RWTS副常式可供使用者利用機器語言從任何軌道和段落上讀取或寫入資料。在運用RWTS 時，須先建立IOB 表（即INPUT／OUTPUT Control Block）及相關的“裝置特性表”。IOB可告訴RWTS 關於磁碟推動機在那一個溝槽、磁碟機號數、所要存取的軌道及段落、以及執行動作（讀或寫）等情報。裝置特性表則提供RWTS一些控制DISKⅡ必要的情報。

IOB 和裝置特性表須先建立在記憶體中。此外，還須寫一個

控制副常式存放在記憶體中。此副常式須經由JSR 指令跳到RWTS 的開始位址（$3D9），此時，在A及Y暫存器中須事先存有IOB 的開始位址（A暫存器中爲高位，Y中爲低位）。IOB的格式如本節末的表3所示。裝置特性表的格式如表4所示。

下面的例子說明如何使用RWTS 副常式。示範性的IOB、裝置特性表、及控制副常式都先要載入到記憶位址$Cφφ之後。

控制副常式將IOB 的開始位址載入A及Y暫存器中，然後跳到RWTS 副常式。如下所示：

```
$C00-   A9 0C       LDA   #$0C   Load A register with $0C
$C02-   A0 0A       LDY   #$0A   Load Y register with $0A
$C04-   20 D9 03    JSR   $03D9  Jump to the RWTS subroutine
$C07-   60          RTS
$C08-   00          BRK
```

下面這個IOB 會存取到溝槽6，磁碟機1，把從$2φφφ 記憶位址開始的256個拜寫入磁碟的軌道18，段落6上去。

Location	Code	Purpose
$C0A	01	IOB type indicator, must be $01
$C0B	60	Slot number times 16
$C0C	01	Disk drive number
$C0D	00	Expected volume number
$C0E	12	Track number
$C0F	06	Sector number
$C10	20	Low-order byte of Device Characteristics Table
$C11	0C	High-order byte of Device Characteristics Table
$C12	00	Low-order byte of data buffer starting address
$C13	20	High-order byte of data buffer starting address
$C14	00	Unused
$C15	00	Unused
$C16	02	Command code, $02 = write
$C17	00	Error Code
$C18	00	Actual volume number
$C19	60	Previous slot number accessed
$C1A	01	Previous drive number accessed

下面這個裝置特性表必須包括在一起，我們把它放在位址，緊接IOB 之後。IOB 的位址$C1φ和$C11指向此裝

置特性表的開始位址。

Location	Code	Purpose
$C2Ø	ØØ	Device type code (put a $ØØ here)
$C21	Ø1	Number of phases per track (put a $Ø1 here)
$C22	EF	Time count (put a $EF here)
$C23	D8	Time count (put a $D8 here)

當IOB 載入$CφA，裝置特性表載入$C2φ，且控制副常式（負責載入A及Y暫存器）的位址在$Cφφ，則

CφφG或

CALL 3φ72

將使此整個常式開始執行。

表3：IOB 的格式

拜	名稱	目的
1	IBTYPE	告訴RWTS 此IOB 的型式。應爲$Ø1。目前尚未定出其他型式。
2	IBSLOT	必須爲SLOT*16的值，此爲磁碟推動機控制板的所在位置。例如，欲存取溝槽6，則此處爲$6Ø。
3	IBDRVN	所欲存取的磁碟機號數—$Ø1或是$Ø2。
4	IBVOL	所欲存取磁碟的册號數。册號$ØØ 可與任何册號符合。
5	IBTRK	所欲存取的軌道（Ø 到34 ）號數，範圍須在$ØØ到$22之間。
6	IBSECT	所欲存取的段落（Ø 到15 ）號數，範圍須在$ØØ到$ØF 之間。
7 & 8	IBDCTP	此二拜須爲裝置特性表的開始位址。拜7

為低位，拜 8 為高位。

9 & 10	IBBUFP	此二拜須為資料緩衝區的開始位址。資料緩衝區為 256 拜長的記憶體，可供 RWTS 運用。若有資料將寫入磁碟，即由此緩衝區寫出。若要從磁碟讀入資料，此資料將先存放在此記憶緩衝區。256 拜是 RWTS 讀寫動作最小且同時是最大的資料長度。
11 & 12	------	未使用。
13	IBCMD	此拜存放 RWTS 所要執行的動作命令碼。可為： $00——虛命令。除使磁碟機起動及讀寫頭歸位之外，不做任何動作。 $01——從指定的軌道及段落上讀取全部 256 個拜，並存放在資料緩衝記憶區中。 $02——將資料緩衝記憶區中的 256 拜資料寫入磁碟中指定的軌道及段落上。 $04——定磁碟的格式。定好格式的磁碟中，每一軌道及段落上都會寫上自同步分塊（ nibble）。因為整個磁碟都重定格式，故第 5 及第 6 拜中的值不需要。整個定好格式的磁碟可即連用，其中無 DOS 或任何資料，有待使用者予以存放。
14	IBSTAT	此拜供執行中發生錯誤時存放錯誤碼。RWTS在跳回時若進位旗號為清除（

clear）狀態，則表示無錯誤發生。若進位旗號爲設定時，此拜即指出所發生錯誤的型式爲何。

$10——磁碟具有寫入保護，不可寫入。

$20——冊號不符合。所找到的磁碟冊號與第4拜中所指定的冊號不同。

$40——磁碟機錯誤。發生某些不正常的現象。

$80——讀取錯誤。RWTS 在經過48次反復嘗試之後，仍然無法讀取到位址欄或資料欄。譬如在某段落的資料欄裏未曾寫入任何資料，則在讀取時就會發生讀取錯誤。

15	IBSMOD	存放實際找到的磁碟冊號數。
16	IOBPSN	存放最近一次存取的溝槽*16值。譬如，剛存取在溝槽5中的磁碟推動器，則此處爲$50。若在指定的溝槽上並無控制器，則磁碟會呆懸（hang）。
17	IOBPDN	存放最近一次存取的磁碟機號數——$01或$02。

表4：裝置特性表的格式

拜	名稱	目的
1	DEVTPC	裝置型式碼，可表示此裝置爲何型式。若爲DISKⅡ，則此拜爲$00。
2	PPTC	應存放$01。

3 & 4	MONTC	馬達開啓時間的互補值，以100微秒（uS）爲間隔單位。若爲DISKⅡ，則第3拜中應爲$FF，第4拜中爲$D8。

第十章

輸入、輸出，及鏈結

選擇輸入/輸出裝置：IN＃,PR＃及其他

Integer BASIC鏈結

Applesoft鏈結

選擇輸入/輸出裝置：IN#，PR#及其他

可以將資料當作輸入或輸出的方法很多。最常用的是以鍵盤當作輸入源，以電視機當作輸出。而任何經過控制器接到溝槽上的週邊附屬裝置都可用IN# 及PR# 命令將它當作是輸入或輸出。

例如：

IN#6　從溝槽6上所接的裝置取得輸入資料。注意：若溝槽6上接磁碟控制器卡片，此命令會使DOS 重新啓動；若溝槽6上未接任何裝置，則可能造成系統懸浮，須按“RESET”鍵重新恢復。

IN#Ø　不從週邊裝置而從鍵盤（非溝槽Ø）上取得輸入資料。

PR#1　將輸出資料傳送到溝槽1所控制的裝置上，通常是列印機。注意：若溝槽1上未裝有任何裝置時，可能造成系統懸浮，須按“RESET”鍵重新恢復。

PR#Ø　將輸出資料送給電視（並非溝槽Ø）。

這些命令的語法爲IN# s或PR# s，其中s爲溝槽號數。根據s值，其可能發生的情況如下：

s值	結果
小於Ø	語法錯誤（SYNTAX ERROR）。
Ø	設定慣用裝置（在IN# 時，爲鍵盤輸入，在PR# 時，爲電視輸出）。
1至7	傳送到該指定溝槽上所控制的裝置（若該溝槽上爲磁碟控制器則先重新啓動DOS）。
8至16	在延遲執行方式時造成語法錯誤；在立即執行方式時系統會懸浮。

17 至 65535 範圍錯誤（ RANGE　ERROR ）。

大於 65535　語法錯誤（ SYNTAX　ERROR ）。

在有DOS 的情況下，IN# 及PR# 命令可按一般的方法（參閱BASIC 手册）用在立即執行方式。但是若用在程式中時，則必須按照DOS 命令的方式，如：

```
10　D$=""：REM　CTRL-D
20　PRINT　D$；"PR#　1"
30　PRINT　D$；"IN#　2"
```

當DOS 不存在時，IN# 和PR# 命令設定APPLE監督程式的輸入及輸出暫存器，以選擇所要的輸入與輸出裝置。

當DOS 存在時，APPLE監督程式的輸入與輸出暫存器被設定以選擇DOS ，而DOS 的輸入與輸出暫存器則被設定以選擇所要的輸入與輸出裝置。以下針對每一個字元進出APPLE所發生的動作加以說明。

當Apple送出一個字元，則Apple 監督程式的輸出暫存器就將此字元送給DOS 。若此字元須轉送（非DOS 命令）時，則DOS 執行以下二步動作：

1. 首先將Apple 監督程式的輸入輸出暫存器內容用DOS 的輸入輸出暫存器取代，然後將此字元送給由現在的監督程式輸入輸出暫存器所選擇的裝置上去。
2. 其次，再將指向 DOS 的指標放入APPLE的輸入輸出暫存器以保持連接。

同理，每當Apple 要求輸入一個字元，監督程式的輸入暫存器就向DOS 提出要求。DOS 接着也執行下列動作：

1. 首先將監督程式中的輸入輸出暫存器內容用DOS 的輸入

輸出暫存器內容取代。然後從這輸入輸出暫存器目前所選擇的裝置上讀取一個輸入字元。

2. 其次，再將指向DOS 的指標放入Apple 的輸入輸出暫存器以保持連接。

當DOS 存在時，DOS 會先從輸入裝置截取到所有即將送到Applesoft 或Integer BASIC或監督程式的輸入字元。這就是爲什麼當鍵入立即執行方式的IN# 或PR# 命令時，DOS 可用以改變DOS 的輸入及輸出暫存器的理由。

同理，DOS 也截取由Apple 送出而尚未抵達輸出裝置的輸出字元（但已改變Apple監督程式的輸入及輸出暫存器）。因此能使程式中送出的IN# 或PR# 命令（經由PRINT方式送出的不算）在送到DOS 之前先將Apple 監督程式的輸入輸出暫存器改變，而使DOS 失去連接。因爲每當DOS 想要收送一個字元時都會取代所有監督程式的輸入輸出暫存器內容，所以當DOS 的輸入輸出暫存器未同時失去連接時，總會將本身連接上。

若是從一程式中送出PR# 命令執行，如：

```
50    PR#1
```

則DOS 會有一陣子失去連接，而不能截取隨後的輸出。但DOS 對輸入仍然保持連接，到下一次輸入字元時，又會使DOS 對輸入與輸出重新保持連接。

對IN# 時情況相同，如下面一行程式

```
60    IN#1
```

會使DOS 對隨後的輸入失去連接，但對輸出仍爲連接。而在下一次要送出字元時（即使是RETURN 或催促符號），會再使DOS 對輸入與輸出保持連接。要防止這種錯失並讓DOS 來掌握輸入及

輸出暫存器，須以立即執行方式來送出PR# 或IN# 命令，或者像前面所說從程式中送出DOS 命令：

```
10   D$ = "":REM  CTRL-D
20   PRINT  D$;  "PR#1"
30   PRINT  D$;  "IN#2"
```

CTRL-D 字元告知DOS：以下的輸出字元爲DOS 命令。

表1：APPLE監督程式的輸入及輸出暫存器

監督程式輸入暫存器：位置爲 56-57 （$38 — $39 ）

設定暫存器是由：	設定值	隨後的輸入從：
RESET	-741	監督程式輸入常式
φ CTRL-K（註1）	（$FD1B）	鍵盤
IN#φ （註2）		
s CTRL-K（註1）	49152 + s*256	溝槽 s
IN#s （註2）	（$Csφφ）	若溝槽 s 上有磁碟
（s > φ）		控制器則啓動DOS
DOS 啓動	-8514+記憶體頂位	DOS
	（-$217E + $記憶體頂位）	

監督程式輸出暫存器：位置爲54-55($36－$37)

設定暫存器是由：	設定值	隨後的輸出到：
RESET	－528	監督程式輸出常式
φ CTRL-P（註1）	($FDFφ)	電視顯像幕
PR#φ （註2）		
s CTRL-P（註1）	45192＋s＊256	溝槽s
PR#s （註2）	($Csφφ)	若溝槽s上有磁碟
(s＞φ)		控制器則啓動DOS
DOS 啓動	－8514＋記憶體頂位	DOS
	(－$2142＋$ 記憶體頂位)	

註1：s CTRL-K和s CTRL-P都是監督程式命令。按CTRL-K的方法是：按住CTRL鍵時再按K鍵。

註2：當DOS 存在時，只有在此命令被當作是程式指令送出且不在PRINT CTRL-D 指令中時，才會影響監督程式暫存器的內容。

註3：除了表1所說的命令之外，亦可用POKE指令直接將適當的值塞進監督程式暫存器中以選擇輸入或輸出裝置，或者將失去連接的DOS 重新連接。

表 2 ：DOS 的輸入及輸出暫存器

DOS 輸入暫存器

設定暫存器是由：	設定值	隨後的輸入從：
DOS 啓動	－741	監督程式輸入常式
RESET 3DφG	（$FD1B）	鍵盤
IN#Ø（註 4 ）		
PRINT D$；“IN#Ø”		
（註 5）		
IN#s（註 4 ）	49152＋s＊256	溝槽 s
PRINT D$；“IN#s”（$Cs ØØ）		
（註 5）		
（s＞Ø）		

DOS 輸出暫存器

設定暫存器是由：	設定值	隨後的輸出列：
DOS 啓動	－528	監督程式輸出常式
RESET 3DØG	（$FDFØ）	電視顯示幕
PR#Ø（註 4 ）		
PRINT D$；“PR#Ø”		
（註 5）		
PR#s（註 4 ）	49152＋s＊256	溝槽 s

設定暫存器是由：	設定值	隨後的輸出列：
PRINT Ds;"PR#s"(	$Cs00)	若溝槽s上有磁碟控制器則重新啓動DOS
(s>0)		

註4：當DOS 存在時，若此命令被當作是程式指令送出且不在PRINT CTRL-D指令中時，則將不會影響到DOS 的輸入及輸出暫存器。

若執行程式中的一行爲

120 PR#3

則監督程式輸出暫存器將會改變，並使DOS 暫時失去連接，直到下一個輸入動作時才又恢復。

註5：此命令中，假設字串變數D$已指定爲CTRL-D 字元。

註6：不管DOS 輸入輸出暫存器所選擇的輸入輸出裝置是什麼，輸入或輸出也可以從磁碟上接受或送出。

註7：除表2所列的命令之外，也可利用POKE 指令直接將適當的值填塞到DOS 的輸入輸出暫存器位址，以選擇輸入輸出裝置。但是，隨着系統記憶體大小及DOS 版本的不同，DOS 輸入輸出暫存器特定的記憶體位置也會變動。因此，須以特別方法將暫存器的內容改變，可分兩步進行：

1. 將監督程式輸入輸出暫存器的位址改變爲所希望的DOS 輸入輸出暫存器位址。（可利用POKE指令直接塞進監督程式暫存器位址，或在程式中執行IN# 或PR# 等非DOS 指令。）
2. CALL1002（在監督程式中，則爲$3EAG）

此處CALL執行之後，DOS 會經由監督程式暫存器重新連

接，而先前的監督程式輸入輸出暫存器內容會出現在DOS的輸入輸出暫存器中。不管程式中須與DOS 暫時失去連接多少次，也都可用此CALL予以再連接。此可參考後面所舉的程式例。

註8：從鍵盤打入的監督程式命令s CTRL-K 或s CTRL-P，直接影響監督程式輸入或輸出暫存器，不被DOS 認知。

Integer BASIC鏈結

有些應用程式可以非常容易的利用一連串程式，先經載入再予執行即可完成。在此種情形下，第二個程式通常需要第一個程式所產生的變數或列陣值。但一般RUN 命令在載入第二程式時會將第一程式的結果值全部抹蓋掉。在Integer BASIC（不是Applesoft）中的DOS 命令CHAIN可用來解決這個問題。

假設有一個Integer BASIC 程式PART ONE 正在使用，則

```
CHAIN PART TWO
```

會載入並執行Integer BASIC 程式PART TWO，此時並不會將PART ONE 程式的任何變數值清洗掉。此一CHAIN命令可用以上所舉的立即執行方式送出，或從PART ONE 程式的最後一行當作DOS 命令予以送出，如下所示：

```
2ØØ1Ø D$=""：REM CTRL-D
2ØØ2Ø PRINT D$；"CHAIN TWO"
```

此命令的語法為：

```
CHAIN f〔,Ss〕〔,Dd〕〔,Vv〕
```

APPLESOFT 鏈結

CHAIN命令只對Integer BASIC 有效，但在不須傳交變數值的情況下，也很容易將Applesoft 程式加以鏈結以便依次的載入執行。在第一個程式的最後一行須爲：

2ØØØ PRINT CHR$(4); "RUN PART TWO"

當執行此行時，將會開動第二個程式（此處爲PART TWO），但也同時將第一程式的結果值全部抹蓋掉。

若想要使Applesoft 程式能有和前面Integer BASIC 相同的鏈結功能，則須用不同的方法。此時須利用DOS System Master磁碟上的一個機器語言程式CHAIN。

要將程式PART ONE和另一程式PART TWO 鏈結起來，首先須使程式CHAIN與程式PART TWO 都在同一磁碟上（見次頁指令）。然後，將下面兩行放入程式PART ONE 的最後兩行：

2ØØØØ PRINT CHR$(4); "BLOAD CHAIN, A52Ø"
2ØØ1Ø CALL 52Ø"PART TWO"

這兩行的行號數可任取，但須使此二行緊接。第一行將Applesoft的鏈結能力載入系統，第二行予以執行（但須注意以下說明）。

在第二行中，CALL 52Ø 和雙引號之間不留間隔，且CALL的位址不可爲十六進位。

若Applesoft 是在ROM 卡（韌體）上，可利用FID 程式（附錄J）將程式CHAIN複製到另一磁碟上。

若爲RAM Applesoft（在磁碟上），除可利用FID 複製外，尚可如下法。首先，利用命令

BLOAD CHAIN, A12296

將CHAIN程式搬進記憶體中；然後，利用命令

BSAVE CHAIN, A12296, L456

，將其保存在所希望的磁碟上。

須注意此處在複製CHAIN時的位址參數完全不和實際使用CHAIN時所用的位址參數相同。

附　錄　A
DOS 命令中用到的幾種檔案型式

檔案型式使用情形（按命令字母順序排列）
檔案型式使用情形（按檔案型式排列）

除非特別指明，否則DOS命令可以用在立即執行方式或延遲執行（放在程式中）方式中。但是，有些主文檔命令（如READ，WRITE）必須用延遲執行方式。

大多數的DOS命令涉及到有名字的檔案。檔案可能是主文（資料）檔，或 Integer BASIC，Applesoft 或機器語言的程式。下面幾個表，指出每一命令可能會用到的檔案型式。第一個表是按命令字母順序排列；第二個表根據其所用檔案的型式而分類排列。至於CATALOG，FP，INT，MAXFILES，MON，NOMON，PR# 和 IN# 等命令因爲沒有明確的用到某一檔案，故此處不予包括在內。

檔案型式使用情形（按命令字母順序排列）

DOS命令所用到的檔案	Integer BASIC 程式檔	Applesoft BASIC 程式檔	順序主文檔	隨機存取主文檔	機器語言二進位檔
APPEND			x		
BLOAD					x
BRUN					x
BSAVE					x
CHAIN	x				
CLOSE			x	x	
DELETE	x	x	x	x	x
EXEC			x		
INIT	x	x			
LOAD	x	x			

DOS命令所用到的檔的檔案	Integer BASIC程式檔	Applesoft BASIC程式檔	循序主文檔	隨機存取主文檔	機器語言二進位檔
LOCK	x	x	x	x	x
OPEN			x	x	
POSITION			x		
READ			x		
RENAME	x	x	x	x	x
RUN	x	x			
SAVE	x	x			
UNLOCK	x	x	x	x	x
VERIFY	x	x	x	x	x
WRITE			x	x	

注意：僅用在延遲執行方式的命令有：APPEND，OPEN，POSITION，PEAD，WRITE

檔案型式使用情形（按檔案型式排列）

只能 Integer BASIC檔

CHAIN

Integer BASIC或Applesoft檔

INIT

LOAD

SAVE

RUN

只能循序主文檔

APPEND

EXEC

POSITION

只能循序主文檔或隨機存取主文檔

OPEN

CLOSE

READ

WRITE

只能機器語言檔

BLOAD

BRUN

BSAVE

各種檔案型式皆可

DELETE

LOCK

UNLOCK

RENAME

VERIFY

注意：只能用在延遲執行方式的命令有：APPEND，OPEN，
POSITION，READ，WRITE

附　錄　B

DOS 訊息

ONERR GOTO碼

討論

當DOS 偵測到磁碟使用上的錯誤時，通常會顯示出說明錯誤原因的訊息並停止任何正在執行中的程式。這些DOS 訊息比一般由Applesoft 或Integer BASIC 所產生的訊息要多。二者之間的差異如下：

Applesoft 的訊息開頭有疑問號，如：

? SYNTAX ERROR

Integer BASIC的訊息開頭有三個星號，如：

*** SYNTAX ERROR

至於DOS 訊息的開頭則無任何符號，如：

SYNTAX ERROR

不管DOS 訊息是由Applesoft ，或Integer BASIC 或監督程式何者送出，內容完全一樣。

若在使用監督程式當中發生DOS 訊息，則系統會Reset 回到原來BASIC 中。

利用Applesoft 的ONERR GOTO指令（參閱Applesoft 手册），可建立一個處理DOS 訊息（此DOS 訊息通常會使程式中斷）的Applesoft 錯誤處理常式。當DOS 錯誤發生在Applesoft程式的ONERR GOTO指令之後時，有關此錯誤的型式碼會被存入記憶器位址222中。此位址也正是Applesoft 存放Applesoft錯誤訊息碼的同一位址。當錯誤發生，造成Applesoft ONERR GOTO跳越到該錯誤處理常式，同時其對應此錯誤的碼可經由指令

Y = PEEK(222)

而成爲Y的值。

DOS 訊息及其對應的Applesoft ONERR GOTO碼如下所

示，並說明其最普通的原因。以下將逐項詳細討論。

ONERR GOTO碼

碼	DOS 訊息	最普通的原因
1	LANGUAGE NOT AVAILABLE	Applesoft 語言不在磁碟上
2、3	RANGE ERROR	命令參數值太大
4	WRITE PROTECTED	磁碟上有寫入保護標記
5	END OF DATA	讀取已結束的主文檔
6	FILE NOT FOUND	檔名拼錯，或不在磁碟上
7	VOLUME MISMATCH	册號不對
8	I／O ERROR	門開著，或磁碟未起始設定
9	DISK FULL	磁碟上檔案太多
10	FILE LOCKED	想把已LOCK（鎖制）過的檔蓋寫掉
11	SYNTAX ERROR	檔名，參數或逗點不對
12	NO BUFFERS AVAILABLE	太多主文檔被開啓（OPEN）
13	FILE TYPE MISMATCH	磁碟檔型與命令不匹配
14	PROGRAM TOO LARGE	可供使用的記憶體不足
15	NOT DIRECT COMMAND	命令須在程式中

討論

LANGUAGE NOT AVAILABLE （ONERR GOTO 碼＝1）

當DOS 找不到Integer BASIC 或Applesoft 語言時產生此訊息（此語言是執行DOS 命令所必須的）。FP ，INT ，LOAD，及RUN 等命令都可造成尋找語言的動作。若所要的是Integer BASIC ，DOS 會在ROM 中找尋。若所要的是Applesoft ，DOS 會首先利用Applesoft ROM 卡上（若有時，且不管卡上的開關位置）的Applesoft ，在ROM 中尋找該語言。若ROM 中找不到，則到自然設定的磁碟推動機（由自然值指定或最近一次使用到的S及D參數值指定）中的磁碟上去尋找。DOS 不會再到其他的磁碟推動機上去尋找。

當DOS 在自然設定的磁碟上找不到Applesoft 程式時，通常即送出此處所說的訊息。此時須更換一片上面有Applesoft 程式的磁碟；或者在任何DOS 命令中使用參數D，以選擇另一磁碟機。像下面這個命令就很適用：

FP ，D2

當你認爲DOS 應該會在ROM 中找到Integer BASIC ，但是並未找到時，可以照以下方法試試：

1. 關掉Apple 電源，並移去蓋板。
2. 在主機板上可找到4個大的ROM 晶片（黑長形）。上面有標明爲“ROM F8 ”，“ROM Fφ ”，“ROM E8 ”，“ROM Eφ ”。
3. 用力按緊這些晶片。
4. 蓋回蓋板，打開電源，再以INT 命令試試。

當你認爲DOS 應該在ROM 卡片（韌體）上找到Applesoft，但是並沒找到時，則可照以下方法試試：

1. 關掉電源，並移去蓋板。
2. 拔下Applesoft ROM 卡。在此卡上可找到5個大的

ROM 晶片，晶片上有標號爲1，2，3，4，5，晶片下則爲Dϕ，D8，Eϕ，E8及Fϕ。

3. 用力按緊這些晶片。
4. 重新將此Applesoft 卡插回溝槽中（即最左邊的）。
5. 蓋回蓋板，打開電源，再以命令FP 試之。

RANGE ERROR （ONERR GOTO碼＝2或3）

當DOS 命令中的數值或參數值太大或太小時，發出此訊號。下面列表說明之。

			範圍	
	參數	字母	最小值	最大值
所有檔案：	溝槽	S	1	7
	磁碟機	D	1	2
	册號	V	ø*	254
循序主文檔：	拜	B	ø	32767
	相對欄位	R	ø	32767
	絕對欄位（EXEC）	R	ø	32767
隨機存取主文檔：	資料錄長度	L	1	32767
	資料錄號數	R	ø	32767
二進位檔：	開始位址	A	ø	65535
	拜數目	L	1	32767

		範圍	
DOS 命令	數值	最小值	最大值
PR# s	s	Ø	16**
IN# s	s	Ø	16**
MAXFILES n	n	1	16

*在 INIT時實際指定的磁碟册號最小值爲 1 。

** 在Apple][中最大溝槽號數爲 7 。只有在延遲執行方式下，當 s值在8到16 時才送SYNTAX ERROR 訊息。

注意：對於超出以上所說範圍的值，並非只有RANGE ERROR的訊息。任何DOS 命令數值或參數值，凡小於Ø或大於 65535 時，將造成SYNTAX ERROR 訊息，而非RANGE ERROR 訊息。

WRITE PROTECTED （ONERR GOTO碼＝4 ）

當DOS 要將資料存入磁碟時，磁碟機在磁碟外殼封套左側偵測不到“寫入保護”的缺口，就會造成此訊息。下面所說的原因最有可能：

1. 在磁碟的“寫入保護”缺口上貼有黏膠帶，用以防止意外蓋寫或刪除掉上面的資料。在DOS 要SAVE或BSAVE或WRITE 資時上去時，必須將此黏帶拿掉。
2. 在磁碟上沒有“寫入保護”的缺口。這是專對System MASTER 磁碟而設，以求最大保障。必要時，也可在此磁碟上切出一個和其他磁碟上大小位置完全相同的缺口來

3. 當正在做COPY（複製）程式時，收到此訊息而其原因並非前面1.2.所述，則很可能是磁碟插入磁碟機不正確（在任何其他情況，DOS 會送I／O ERROR訊息以指示磁碟插入不正確）。檢查磁碟在磁碟機中的位置，並再參閱第一章對插入磁碟的說明。

END OF DATA（ONERR GOTO碼＝5）

當想要從一個主文檔中並未存有資料的部分取出資料時，就造成此訊息。任何在循序主文檔最後欄位之外，或隨機存取主文檔每一資料錄最後欄位之外的拜都可能包含Ø值。在ASCII 碼中零就是"空白"字元，任何命令在執行中取到此字元時，就會造成END OF DATA訊息。只有OPEN會自動將檔中位置指標設定回到檔的開始。此訊息經常發生在INPUT 或GET 命令之後，發生的原因可能是：

1. 連續的INPUT 太多，或INPUT 中有太多的變數。因爲每一個INPUT 或INPUT 變數都會往下讀取檔中的一欄資料。
2. 連續的GET 太多。每一個GET 都會往下讀取一拜或一字元。
3. 參數B（表示Byte ）太長。在循序檔中，B參數絕不可指定超出檔中最後RETURN 字元以後的拜。在隨機存取檔中，B參數不應指定超出目前所選資料錄中最後RETURN 字元以後的拜。須記得在檔或資料錄中的第一個拜是拜Ø。
4. 在命令POSITION 中的R（代表相對欄位）參數太大。

在循序檔中，此參數絕不可指定超出檔中最後一個資料欄以外的資料欄。在隨機存取檔中，R參數不應指定超出目前所選資料欄中最後一個資料欄以外的資料欄。須記住，POSITION中的R參數與READ中的R參數不同。此處的R參數指定檔中相對於目前在檔中位置的欄位，且僅向前進位。R0 表示位置不變。R1 將檔中位置往前跳到目前資料欄所在位置之後的第一個拜。

POSITION往前逐拜地掃瞄檔案內容，以找尋第Rp 個RETURN 字元。若在找到所要的RETURN 字元之前，碰到一個0拜（“空無”字元），則立即造成END OF DATA訊息：即實際上並不一定是INPUT 或GET 時讀到“空無”字元才會造成此訊息。

5. 在EXEC命令中R參數（代表絕對欄位）太長。此參數可以指定檔中最後一欄之後的第一欄，但若指定檔案結束之後的第二欄則造成END OF DATA訊息。記住，R0 指定檔中的第一欄。

6. 在READ命令中R參數（代表Record）指定隨機存取檔中一筆未經存入的資料錄。在讀取隨機存取檔中某一特定資料之前，該筆資料必須事先已寫入。

須記住：READ的R參數與POSITION 或EXEC所用的R參數不同。READ中的R參數指定檔中一筆絕對位置上的資料，R0 爲第一筆資料錄。

DOS 利用命令OPEN中的L參數值來計算第Rr 筆資料錄的開始，因此，在做READ之前的OPEN中的L參數值必須使用從前WRITE 時OPEN所用的同一L值。

FILE NOT FOUND （ONERR GOTO 碼=6）

當DOS 命令中指定的檔名在所選擇的或自然設定的磁碟推動機中的磁碟目錄（ CATALOG ）上找不到時，就造成此訊息。只有SAVE，BSAVE ，INIT，和OPEN等命令可以新建一個原來不存在的檔案。此外，CLOSE 可用在任何有效的名稱。被其他任何DOS 命令所指定的檔名必須是已存在於磁碟上方可。

此訊息發生的原因爲：

1. 可能拼錯或打錯檔案名稱，或漏掉一個用來分隔參數的逗點。檢查CATALOG以找出正確的拼法。警告：如果意外地在檔名中打入控制字元，CATALOG並不會將該字元顯示出來。關於此點，請參閱附錄F“檔案名稱”。
2. 此檔在另一磁碟上。檢查CATALOG。
3. 此檔已被意外地刪除掉。檢查CATALOG。
4. 對某一磁碟執行INIT命令，其中有一個檔名是每次在利用該磁碟（在磁碟機1號上）啓動系統後DOS 接下來就要執行的。除非另寫一個BASIC 程式存入，名字是INIT命令中所用的，否則每次利用該磁碟在磁碟機1號上啓動系統時，都會產生FILE NOT FOUND訊息。如果已經不記得此一「招呼程式」的名稱，則必須對此磁碟重作INIT，或者是執行MASTER CREATE 在不致破壞磁碟內容的情況下重新爲招呼（ GREETING ）程式取名字。

册號不符 （ ONERR GOTO碼＝7 ）

當DOS 命令中的V（ 册號 ）參數與磁碟上當起始設定所指

定的册號不同時即產生此訊息。磁碟的册號在每一CATALOG顯示的開頭。除非DOS 命令中特別指定册號，一般都對其省略不用，且無訊息產生。若指定册號Ø，也仍然是將册號省略不用。若在INIT時不指定册號，或指定爲Ø，則磁碟的册號數會被指定爲自然值254。

I／O ERROR（ ONERR GOTO碼＝8 ）

當在磁碟上存取資料失敗（ DOS 會重覆嘗試96 次才放棄）後即造成此訊息。其發生原因有：

1. 所選到的或自然設定的磁碟機拉門未關。須將門關好。
2. 所選到的或自然設定的磁碟機中沒有磁碟。須放進一片磁碟並將門關好。
3. 在所選到的或自然設定的磁碟機中的磁碟未被起始設定過。將該磁碟 INIT。
4. 磁碟插得不正確。檢查磁碟，再參閱第一章對插入磁碟的說明。
5. 在執行VERIFY （查證）命令時，DOS 發現所指定的檔案在磁碟上儲存不正確。若檔案的資料仍在記憶體中，則再嘗試存入（也許存入另一磁碟上）。
6. DOS 命令中的D（磁碟機）參數所指定的磁碟推動機在系統中不存在或自然設定的磁碟機目前是不存的。在下一個DOS 命令中指定正確的D參數可重新設定自然值。
7. 使用16 段落式的DOS 來存取13 段落式的磁碟。利用MUFFIN 程式將磁碟更正爲16 段落式。
8. DOS 命令中的S（溝槽）參數所指定的溝槽上並沒有一

個磁碟控制器。

碰到此困難不要灰心，仍然有救。問題發生在目前的自然設定值是上一個DOS 命令所指定的空溝槽。若下一個DOS 命令中未指定溝槽參數，則仍然會造成相同訊息。重新正確設定溝槽的方法如下：

1. 按RESET 鍵
2. 若出現監督程式催促符號＊ 則鍵入3DØG
3. 鍵入CATALOG Ss ，其中s爲正確的溝槽。

DISK FULL （ONERR GOTO碼＝9 ）

DOS 要在磁碟上存資料時，發現磁碟上已無可用的儲存空間時造成此訊息。使用者可用以存資料的段落數，可從CATALOG顯示上看出（若一檔佔用的段落數超過255個，則CATALOG的顯示又會從ØØØ開始）。若收到DISK FULL 的訊息，稍待以使DOS 將所有可能的資料保存後關閉檔案（目前處理的檔案有部分未存入磁碟）。譬如，當你正在將檔案STUFF 存入磁碟而收到此訊息時，你首先應該

DELETE STUFF

然後將你的程式存入另一個空位較多的磁碟上。

若收到DISK FULL 訊息，然後在刪除任何檔案前緊接著就要SAVE, BSAVE ，或WRITE 任何檔案到磁碟上，則在CATALOG上顯示的進入點中，其中一個的段落長度會變成Ø。不要絕望；因爲儘管在CATALOG顯示上不對勁，但檔案本身卻仍然是完好的。此種情形可以在收到DISK FULL 訊息時，先刪除一些檔

案之後再存入檔案即可避免。

FILE LOCKED (ONERR GOTO碼=10)

在SAVE，BSAVE，WRITE，或DELETE時使用的檔案名稱，在所選擇或自然設定的磁碟推動機中的磁碟上已被鎖制住，即造成此訊息。檢查CATALOG顯示：被鎖制的檔名之前都有*記號。有鎖住的檔案可防止意外的蓋寫。使用另一片磁碟或將所要的檔案解制(UNLOCK)。

SYNTAX ERROR (ONERR GOTO碼=11)

DOS 在DOS 命令中碰到語法錯誤時造成此訊息。翻查手册以找出該命令的正確語法。問題也許出在不合的檔案名稱(見附錄F)，不正確的參數符號，遺漏了參數，或分隔符號(通常是逗點)錯誤或漏失。參數值或命令中的數目值小於0或大於65535時也會造成此訊息。

每一個DOS 命令都造成Applesoft 或Integer BASIC上的Syntax Error，這種情形是很少碰到的。這通常意謂DOS未被啓動(booted)或對輸入輸出已呈不連接的狀態。可試按RESET 鍵，然後鍵入3D0G以重新連接上DOS，或重新啓動磁碟。

NO BUFFERS AVAILABLE (ONERR GOTO碼=12)

當DOS 命令需要更多輸入輸出檔案緩衝區，而所有可用的檔

案緩衝區卻已用完時，造成此訊息。在啓動系統時，DOS 會在記憶器中保留足夠的空間供三個輸入輸出檔案緩衝區之用。後來的MAXFILES命令可以增加或減少檔案的可用緩衝區個數。而命令CLOSE 可釋放主文檔目前佔用的檔案緩衝區。

許多DOS 命令在執行時使用一個檔案緩衝區供輸入或輸出之用，而在停止執行時交回該緩衝區。

當開啓（OPEN）主文檔時，會指派一個檔案緩衝區給該檔供其作輸入及輸出之用。通常，此一檔案緩衝區一直在使用狀態直到該檔被指名關閉或被不指名的CLOSE 命令將所有主文檔關閉以收回緩衝區爲止。當程式結束時並不自動關閉檔案。爲保護緩衝區的目用量，應在檔案用好之後立即關閉。但須記住下一次開啓時，檔中位置指標會重新設定爲檔案的開頭。

在寫入或載入程式之前，可用MAXFILES命令增加緩衝區數目。在MAXFILES中增加時，會使HIMEN往下移，而抹蓋掉已存入的BASIC 程式或Applesoft 字串。在程式的中間改變MAXFILES尤其危險。

FILE TYPE MISMATCH （ONERR GOTO碼＝13）

當DOS 命令使用的檔名已經指定給某一檔案，而該檔的型式不符合此一命令時造成此訊息。若此命令正確無誤，則須改一個檔案名稱，或換一片磁碟，或將已有的檔案刪除或者改名（RENAME）。

DOS 命令與已存在的檔案型式之間不正確的組合會造成幾種不同的訊息。下面是幾種正確的組合：

LOAD f,RUN f,SAVE f	f必須是Applesoft 或 Integer BASIC 程式檔。
CHAIN f	f必須是 Integer BASIC 程式檔。
OPEN f,READ f,WRITE f,	f必須是主文檔。
APPEND f,POSITION f,	
EXEC f	
BLOAD f,BRUN f,BSAVE f	f必須是二進位程式或資料檔。

在INIT或UPDATE 命令中所用招呼程式的檔名必須指到一個Applesoft 或Integer BASIC 程式檔。

PROGRAM TOO LARGE (ONERR GOTO碼＝14)

DOS 命令要將記憶體中的檔案存放到磁碟上時，發現磁碟上的空間不足以完全包含此檔，即造成此訊息。此時可能HIMEM設得太低，或MAXFILES中指定值過大，而使HIMEN太低。欲將檔案緩衝區的數目設爲3個，可用以下命令：

MAXFILES 3

則HIMEN會回到附錄D表2中所述的啓動值。

若正在使用Integer BASIC ，而HIMEN設定很低（譬如，爲了保護高解析圖形記憶體），則在搬移到磁碟Applesoft上時可能會有困難。磁碟Applesoft 佔用約12.5 K的記憶體，但移到磁碟Applesoft （利用FP或LOAD，或RUN ）時並不

重新設定HIMEN爲最大值。當DOS 要從磁碟上載入Applesoft程式時，若HIMEN在約131*00* 之下時會造成PROGRAM TOO LARGE 訊息。參閱附錄D表2 。

DOS 只檢查此程式所佔用的磁碟段落數，即可決定此程式是否裝得進可用的記憶體中。通常，程式不會完全佔滿最後一個段落（256拜），但DOS 忽略此事。DOS 只將LOMEN（Integer BASIC ）或HIMEN（Applesoft ）的高位拜拿來和要裝入的程式結束位置的高位拜比較。因此，一個可以裝入記憶體的程式，但在載入後留下小於256拜的空間記憶位置，可能造成PROGRAM TOO LARGE 的訊息。有時可以在載入此程式之前稍爲移動HIMEN或LOMEN，以改變高位拜，加以改正。

NOT DIRECT COMMAND （ONERR GOTO碼＝15 ）

以立即執行方式執行主文檔命令APPEND ，OPEN，POSITION ，READ或WRITE時產生此訊息。這些命令只能在程式中以PRINT 指令方式使用。

附　錄　C
磁碟資料的格式

儲存過程的簡要說明
寫入循序主文檔
寫入隨機存取主文檔
DOS如何寫入主文檔：一般過程
檔案段落的內容
T/S 表（軌道/ 段落表）
磁碟目錄表
VTOC （磁碟內容表）
軌道比次對映
軌道和段落的配置順序
從磁碟取出資料
從循序檔讀出
從隨機存取檔讀出

此一部分介紹在磁碟上如何儲存資料，以及DOS如何記住某一特定資料的儲存位置。

在下面的討論中，若一數目前附有$符號時即表示該數爲十六進位。

儲存過程的簡要說明

在DISKⅡ系統中，資料是記錄在一個劃分爲 35 個同心圓（稱爲軌道）的磁碟上。這些軌道由最外層的$00算起，到最內層的$22爲止。推動器的讀寫磁頭可以進出運動，停在旋轉磁碟上 35 個軌道中的任一軌道上。

深一層看，每一軌道又分爲 16 節，稱爲段落（Sector）。由$0編號到$F，在每一段落中有256個拜。當讀寫頭固定在某一軌道上時，其中的 16 個段落會隨著磁碟的旋轉而一一通過讀寫頭。DOS每次都以256拜爲單位（亦即一整個段落）將資料寫入磁碟。

DOS儲存資料到磁碟時首先將256拜的資料（一個段落）存放在一塊叫做檔案緩衝區的記憶器中。存滿後即寫入磁碟的一個段落中，然後再存放其次的256拜到緩衝區，乃至磁碟中。

通常，只要DOS能找到一個未經使用的段落即可開始儲存程式或主文檔。當此段落寫滿256拜資料之後，DOS會再找一個空閒段落（也許此段落在另一磁軌上）來繼續儲存資料。這種過程會繼續進行直到整個檔案存放完畢爲止。

爲了能夠記住某一檔案的資料存放在那一軌道，那一段落上，DOS在存放檔案時，也同時將其所用到的軌道及段落記錄在一表中，這個表叫做T／S表（軌道段落表），存放在磁碟上其他的段

落中。

最後，檔案名稱、型式、段落長度以及檔案T／S表在磁碟上的位置等都記錄在軌道$11中一塊叫做目錄表的特別區域中。同時，磁碟的軌道比次對映也予以更新，以正確的反應每一軌道上目前段落的使用情形。

寫入循序主文檔

一個主文檔的進入點包含1到32767個以ASCII碼形式存放的字元，而以RETURN 字元爲結束。（此字元可爲ASCII中的$ØD或$8D）。每一個進入點稱爲一欄。

在一循序主文檔（即OPEN時未指定長度）中，欄與欄之間在儲存時是互相緊接的（見第六章）。DOS會在前一欄結束的RETURN 字元之後緊接著就寫入下一欄的第一個拜（除非命令中有指定Byte 參數）。在每次OPEN此檔時，DOS會忘掉其在檔中目前的位置，而又重新由拜Ø開始寫入（除非命令中有用到B（Byte ）參數）。

若想重寫循序檔中的某一特別資料欄或字元時，可在WRITE命令中使用B（即Byte) 參數，即可由此檔中絕對拜位上的特定拜開始寫入（檔中以拜ϕ爲第一個拜）。此拜可在檔中目前位置之前或之後。

由於很不容易準確的記住那一個字元是主文檔（尤其是循序主文檔）中的那一個拜，所以在循序主文檔中建議最好是不要使用B參數。

POSITION 命令中可使用R參數（相對欄位）將檔案指標從

目前在檔案中的位置前進 r 個欄位。如下面一段程式所示：

```
120  PRINT D$; "OPEN NAMES"
130  PRINT D$; "POSITION NAMES, R1
     3"
140  PRINT D$; "WRITE NAMES"
160  PRINT "APPLE COMPUTER"
170  PRINT D$; "CLOSE NAMES"
```

將嘗試把 " APPLE COMPUTER " 一串字元寫入檔案 NAMES 由第 14 欄（第 1 欄是 R ϕ ）第 1 個拜開始的位置中。

POSITION 可移動到相對於目前循序主文檔位置的任何欄位的第一個拜。若重寫該欄，必須確使所重新PRINT 出來的字數與原來 PRINT 的字數完全相同。若 PRINT 的字數比以前的少，則會產生兩個新欄：一是剛 PRINT 出來的，一是原來欄中的尾端部分。若 PRINT 的字數比以前的多，則會蓋寫掉下一欄開頭部分的字元。

寫入隨機存取主文檔

隨機存取主文檔在開檔（ OPEN ）之時，就須指定 L （ Length ）參數。L 參數決定在一筆資料錄中拜的數目。資料錄由一欄或多欄集合而成，是 DOS 處理的單位。在隨機存取主文檔中每一筆資料錄就像一個分離的循序主文檔，其最大長度是由 L 參數指定。只要在此最大長度內，任何寫入或重寫，却不會影響到其他的資料錄。WRITE 中可使用 R （ Record ）及 B （ Byte) 參數以便指定資料錄及其中的任一個拜，作爲寫入的開始。

因爲任何DOS命令都會終止WRITE 動作，故不能使用POSITION 命令來將WRITE 所指定的資料錄中的欄位往前移動。

DOS利用L參數來計算在每一新資料錄中寫入第一個拜的位置（在前一資料錄第一拜之後的L個拜）。DOS會跳過前一筆資料錄最後一個字元與拜L之間的任何拜。被跳過的拜原來所包含的值會仍然保持不變（細節請看下一節）。

若在隨機存取資料錄中寫入比L參數所指定的值還多的字元，則所有的字元會正確的存入磁碟。但是，當開始寫入下一筆資料錄時，DOS計算新資料錄的開始位置會把前一筆資料錄當作在此指定長度內。因此，新資料錄會蓋寫掉前一過長資料錄最後部分的字元，包括最後的結束記號RETURN 字元。結果是混亂不堪。

DOS如何寫入主文檔：一般過程

當WRITE 一欄到主文檔時，DOS首先檢查磁碟上應該包含該欄的段落中是否已存有資料。若檔案以前從未使用過該段落，則DOS會將ϕ放入Apple檔案緩衝區的全部256個拜中，然後，該緩衝器中可存放資料，以便存入正確的磁碟段落中。當存在緩衝區中256拜的資料裝滿之後，或檔案關閉時（CLOSE），緩衝區的內容即存入磁碟。

因此，當第一次要寫入某一段落時，未使用的拜中皆爲ϕ值。讀取時，碰到包含ϕ值的拜將造成END OF DATA 訊息。

但若DOS發現檔案已在該段落中存有資料，則將該段落中所有的256個拜讀進Apple檔案緩衝區中，此時可在檔案緩衝區中將某些拜改爲所欲寫入的新資料，然後DOS會將此緩衝區的內容再

存回原來讀出的磁碟段落中。WRITE，POSITION（只對循序檔）和PRINT 命令都會如此動作。當所要改變的拜不在目前已被讀進檔案緩衝區中的段落上時，或當檔案關閉時，檔案緩衝區的內容就存回磁碟。

因此，若在一檔中寫入較多資料，且DOS將這些資料儲存在已被此檔使用過的段落中時，將不會對未使用的拜重寫入任何零進去。任何未重寫段落中的拜將繼續保持其原來的值。對循序主文檔結束部分未使用的拜是如此，對隨機存取主文檔每一固定長度資料錄中未使用的拜也是如此。

檔案段落的內容

對存錄一個檔案到磁碟上的一般過程解釋過之後，即可對每一步驟進一步說明。實際儲存的資料，在每個檔案型式的段落中都是不同的。

對不同檔案型式的段落格式

檔案型式	段落	拜（十六進制）	拜的內容
BASIC	第1個段落	0	程式長度，低位
（二種型式）		1	程式長度，高位
		2到FF	字符化（Tokenized）的程式
	以後的段落所有的拜		字符化（Tokenized）的程式

檔案型式	段　落	拜（十六進制）	拜　的　內　容
主　文　檔	所有段落	所有的拜	ASCII碼表示：1拜／字元 （$ØØ表示檔案結束）
二　進　制	第1個段落	Ø	開始的RAM位址，低位
		1	開始的RAM位址，高位
		2	RAM映像長度，低位
		3	RAM映像長度，高位
		4到FF	二進位資料
	以後的段落	所有的拜	二進位資料

T／S　（軌道／段落表）

當一個檔案儲存在磁碟上時，DOS建立起此檔使用磁碟的位置表。此T／S表也照檔案儲存方式一樣存在磁碟上。T／S表的內容如下：

T／S表的第1個段落

拜（十六進制）	拜的內容
Ø	未使用
1	鏈結：T／S表的延續軌道號碼
2	鏈結：T／S表的延續段落號碼 （若鏈結拜皆爲Ø，則爲無鏈結）
3到4	未使用
5—6	段落基準號碼

拜（十六進位）	拜的內容
7 — B	未使用
C	第 1 個檔案段落的軌道號碼
D	第 1 個檔案段落的段落號碼
E	第 2 個檔案段落的軌道號碼
F	第 2 個檔案段落的段落號碼
10	第 3 個檔案段落的軌道號碼
11	第 3 個檔案段落的段落號碼
⋮ ⋮	⋮ ⋮ ⋮ ⋮
F E	第 122 個檔案段落的軌道號碼
F F	第 122 個檔案段落的段落號碼

若任何軌道／段落的值是$\not{0}$／$\not{0}$，表示這是未經過指定的段落（通常也表示檔案結束）。

T／S表接下來的段落（若其擴展超過 122 個軌道／段落對）與上面所說的第 1 個段落大致相同，只是在指向下一群 122 個檔案段落的軌道／段落對有差別。而每一相連段落的鏈結拜 1 及 2也有不同。每一對鏈結拜可讓DOS知道此T／S表下一部分在磁碟上的位置。若此兩個拜都爲$\not{0}$，即表示已到了T／S表的最後部分

主文檔T／S表中不爲$\not{0}$的軌道／段落對才表示該段落中實際存有資料。DOS根據T／S表中的值來計算正確的位置，未指定的軌道／段落對則填$\not{0}$。

因此，若一隨機存取檔的L參數是 128（每一段落兩筆資料錄），而僅寫入一筆資料錄，其號數爲 27$\not{0}\not{0}$，則須使用 13 個段落：其中一個存資料錄 27$\not{0}\not{0}$的內容，另外十二個則存放T／S表。資料錄$\not{0}$到 2683 將來可能會佔用 1342 個段落，但在未寫入之前，並不會佔用磁碟的空間。T／S表指出包含有資料錄號數$\not{0}$到

2683 的十一段落的位置。

磁碟目錄表

在每一片經過起始設定後的磁碟上，軌道$11是被用來存放有關磁碟內容的資料。在此目錄表中，DOS將每一個檔案的名稱，型式，佔用的段落數（MOD 256 —除以256所得的餘數）及其T／S表的位置。命令 CATALOG可將這些資料的大部分顯示出來。目錄表各段落的格式如下：

磁碟目錄表的一個段落

拜（十六進制）	內　容
0	未使用
1	鏈結：目錄表次一段落的軌道號數（通常爲$11
2	鏈結：目錄表次一段落的段落號數 （若兩個鏈結拜皆爲0，則表示結束）
3到A	未使用
B到2D	檔案1的目錄表進入點（見下文）
2E到50	檔案2的目錄表進入點
51到73	檔案3的目錄表進入點
74到96	檔案4的目錄表進入點
97到B9	檔案5的目錄表進入點
BA到DC	檔案6的目錄表進入點
DD到FF	檔案7的目錄表進入點

此處所舉七個目錄表進入點的檔案號碼是任選的。當一個檔案被刪除（DELETE）掉，DOS會將其目錄表進入點標示（見下表）。到以後另一檔案存入時，DOS會將此有標示的舊進入點用新

檔的進入點取代。因此，雖然ＤＯＳ原來很有順序的填入目錄表，但一有檔案刪除，就會改變順序。

磁碟目錄表從軌道$11，段落$Ｆ開始。如果需要更多的目錄表進入點，則段落$Ｆ鏈結到段落$Ｅ，若還要，則段落$Ｅ鏈結到$Ｄ，依此，一直到$１。因此，目錄表最多可存放105個不同檔案的進入點。

每一目錄表進入點的格式如下：

一個檔案的目錄表進入點

相對拜位（十六進制）	內　　容
0	檔案的Ｔ／Ｓ表軌道號數（當檔案被刪除時，此值複製到 20 中，而將拜 0 中的值改爲 $ＦＦ）
1	檔案的Ｔ／Ｓ表段落號數
2	檔案型式（見下面說明）
3 到 20	檔案名稱
21 — 22	段落數目：此檔佔用的磁碟段落數（MOD 256）

此處以相對拜位說明，是因爲在一目錄表段落中，每一個進入點實際開始的拜位不相同。欲找出一相對拜位在段落中的絕對拜位，只要將其相對拜位加上所在進入點第一個拜在段落中的絕對拜位即可（上表已列出）。

因爲只用一個拜存放檔案佔用的段落數，故最大值爲255（$ＦＦ）。若佔用的段落超過255個，其段落數目又從000開始算起（由CATALOG顯示出來）。雖然不致影響到檔案的使用，但可能會對磁碟使用量產生誤解。

檔案進入表中第二個相對拜位指出檔案型式，其八個比次的指定值意義如下：

指示檔案型式的拜

比次	目錄符號	所指示的檔案型式
7	*	此比次爲1時，表示檔案已鎖制（有寫入保護）
		此比次爲0時表示檔案未鎖制（無寫入保護）
6		將來型式擴充時使用（通常爲0）
5		將來型式擴充時使用（通常爲0）
4		將來型式擴充時使用（通常爲0）
3		將來型式擴充時使用（通常爲0）
2	B	此比次爲1時，表示此爲二進制檔
1	A	此比次爲1時，表示此爲Applesoft BASIC 檔
0	I	此比次爲1時，表示此爲 Integer BASIC 檔
	T	比次0到6皆爲0時，表示此爲主文檔

檔案型式由比次0到6中出現的1來決定。若0到6全爲0，則自然設定此爲主文檔。

因此，指示檔案型式的拜可表示爲以下的值：

· 檔案型式拜的值（十六進位）

檔案型式	檔案未鎖制	檔案有鎖制
主文檔	0	8 0
Integer	1	8 1
Applesoft	2	8 2
二進位	4	8 4

VTOC（磁碟內容表）

磁軌$11的段落$0包含有磁碟的內容表，簡稱VTOC。其中

所存放的資料如下：

拜（十六進制）	值（十六進制）	說　　明
0	2	未使用
1	11	第一個目錄表段落的軌道號數
2	0F	第一個目錄表段落的段落號數
3	4	DOS發行號數
4 — 5	0	未使用
6	1到FE	磁碟册號數（自然值：$FE）
7 — 26	0	未使用
27	7A	T／S表中每一段落最多可能的T／S對
28 — 2F	0	未使用
30	FF	此四拜爲軌道比次對映的面罩
31	FF	（見後二頁）：比次爲1時表
32	00	示該段落在使用中
33	00	
34	23	磁碟中的軌道數。
35	0F	每一軌道中的段落數。
36	00	每一段落的拜數（低位）。
37	01	每一段落的拜數（高位）。
38 — 3B	0	軌道0的比次對映　（這些軌
3C — 3F	0	軌道1的比次對映　道可供使
40 — 43	0	軌道2的比次對映　用者運用）
44 & 45	?	軌道3的比次對映
46 & 47	0	軌道3的比次對映
48 & 49	?	軌道4的比次對映

拜（十六進制）	值（十六進制）	說　　明
4 A & 4 B	0	軌道 4 的比次對應
⋮	⋮	⋮
⋮	⋮	⋮
78 & 79	?	軌道 $10的比次對應
7 A & 7 B	0	軌道 $10的比次對映
7 C — 7 F	0	軌道 $11的比次對映（目錄表及 VTOC ）
8 0 & 81	?	軌道 $12的比次對映
82 和 83	0	軌道 $12的比次對映
⋮		
⋮		
C 0 及 C 1	?	軌道 $22的比次對映
C 2 & C 3	0	軌道 $22 的比次對映
C 4 — F F	0	未使用

軌道比次對映

VTOC的拜從 $38開始，每四個連續的拜組構成了一條軌道的比次對映。其中0與 1 的排列表示該軌道中的各段落目前何者在使用中或是空閒未用。其格式如下：

一條軌道的比次對映

拜	比次	所指定的段落（十六進制）	拜	比次	所指定的段落（十六進制）
1	7	F	2	7	7
	6	E		6	6
	5	D		5	5
	4	C		4	4
	3	B		3	3
	2	A		2	2
	1	9		1	1
	Ø	8		Ø	Ø

－－－－－－－－－－－－－－－－－－－－－－－－

拜 3 及拜 4 全部爲備用

軌道比次對映中的比次值若爲 1 ，表示其對應的段落空閒未用。反之，若爲 Ø ，則表示目前在使用中。上表中標示爲“備用”的比次皆未使用，其值均爲 Ø 。一條典型軌道的比次對映如下所示：

第一個拜	第二個拜	第三個拜	第四個拜
ØØØØØØ11	11111111	ØØØØØØØØ	ØØØØØØØØ
↑↑↑↑↑↑↑↑	↑↑↑↑↑↑↑↑	備　用	備　用
FEDCBA98	7654321Ø		
所指定的段落		未　使　用	

1 ＝空閒未用的段落

Ø＝使用中的段落

當一個檔案存入磁碟（利用WRITE，SAVE，或BSAVE）時，一條完整的磁軌即配置給此檔（可能時），此時軌道比次對映將整條軌道標示爲使用中。當此檔關閉（CLOSE）時，比次對映中再將實際未使用到的段落標示爲空閒未用。

當某一檔名被刪除時，其實際存有檔案資料的段落只能被設定爲空閒未用。譬如說，你的磁碟上放有一個佔了100個段落的BASIC檔，叫做BIG。如果現在以同一名字BIG在同一磁碟上存入一個佔2個段落的檔案（蓋寫原來的），則在目錄（CATALOG）上會顯示你那2個段落的檔案BIG仍然佔住100個段落，利用下面的命令，可以釋放所有BIG中用不到的段落：

```
LOAD BIG
DELETE BIG
SAVE BIG
```

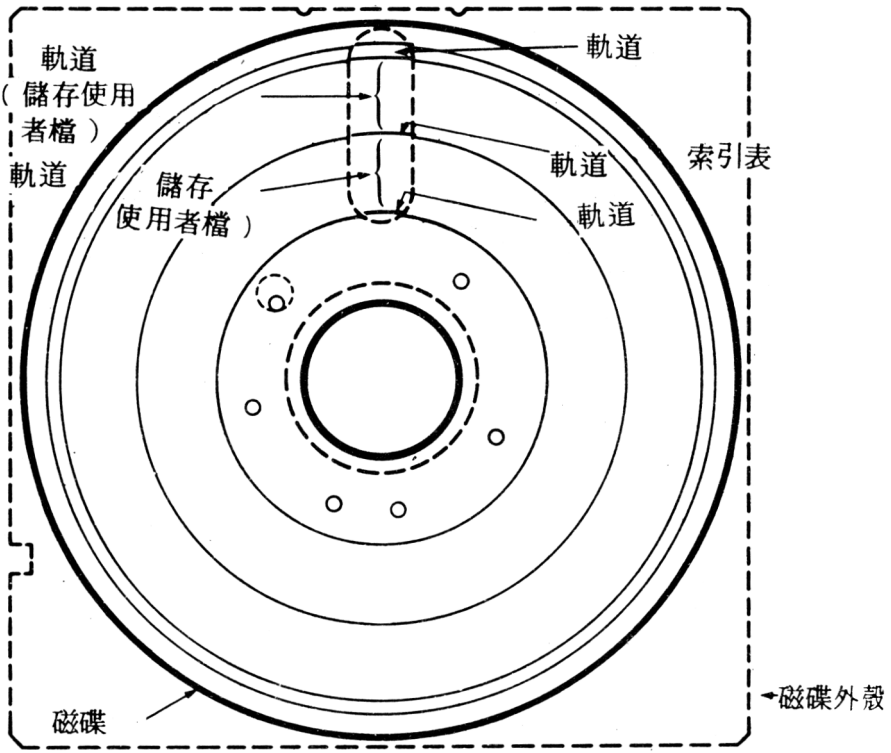
對二進位檔也可用同樣的方法來釋放不需用到的段落。

要釋放主文檔中不需要的段落，必須先將其每一資料欄讀入記憶器中。若所有資料都已保存在一列陣中，則可在使用原來的檔名把每一筆資料寫回磁碟之前，將原來的檔案刪除。另外一個方法是將其每一資料欄讀入Apple之後，再用另外一個檔名寫入磁碟。讀寫完成之後，就可將原來的檔案刪除。

軌道和段落的配置順序

每一個磁碟有35條軌道，其中三條保留給DOS，一條保留給目錄表，剩下31條給使用者運用。每一軌道包含16個段落，所以總共有31 × 16 = 496個段落可用。

使用段落時，是先從段落$ F用起，最後才到段落$ Ø 。軌道則是先從軌道$12開始用起（緊接目錄表／VTOC軌道內側），繼續往內直到軌道 $22（最內層軌道）。當軌道 $22用滿之後，再從軌道$1Ø開始（緊接目錄表／VTOC軌道外側），繼續往外直到軌道$ 3（使用者可使用的最外層）。



軌道配置順序

	最先填		最後填
先	$12	--->	$22
後	$1Ø	--->	$Ø3

段落配置順序

最先填		最後填
$F	--->	$Ø

從磁碟取出資料

DOS用儲存檔案的相反程序從磁碟上取出檔案。譬如，當執行了下面的命令：

```
LOAD FILE　或
BLOAD FILE
```

之後，DOS找到軌道$11上的磁碟檔案目錄表，並找出含有名字FILE的目錄表進入點。此進入點也包含有該檔T／S表的磁碟位置（以軌道，段落表示）。然後，DOS跳到此T／S表讀取第一個T／S對。此值指定該檔第一個段落的磁碟位置。當DOS將該檔第一個段落讀進Apple之後，再回到T／S表找出第二個段落的位置，同樣繼續執行。

從循序檔讀取出

從循序主文檔讀出的過程與載入時很像，譬如下面一段程式

```
50 PRINT D$; "READ TEXT FILE"
60 INPUT A$
```

但是，當執行INPUT時，只有包含了主文檔下一個資料欄（從檔中現在位址到下一個RETURN字元之間的所有字元）的段落才被讀進Apple的檔案緩衝區中。然後，將組成所要資料欄的實際拜位指定爲變數A$。只要資料欄的範圍超過一個段落，就要重覆這個過程。每一個INPUT指令都會恢復讀取動作，也許是從已有適當資料的檔案緩衝區，也許是從磁碟上讀取另一個段落。此一過程將繼讀到檔案結束或關閉爲止。

在READ中使用B（Byte）參數，可使下一個INPUT動作從

指定的絕對拜位（第一個拜爲0，第二拜爲1等）開始讀取。此一拜位可在目前檔中位置之前或之後。要有效的使用此參數，必須對檔中每一拜的內容相當瞭解才行。在POSITION命令中使用R（相對欄位）參數，可將DOS的現在位置指標前進（只能前進）R個欄位。每次開啓檔案，DOS都會忘記原來在檔中的位置，而再重新從檔案開頭讀取（除非另有指定B參數）。

在Integer BASIC與Applesoft中的INPUT指令處理輸入的方式略有不同。若有冒號或逗點字元出現在輸入欄中時，該字元以後的字元可能被放棄或指定到INPUT中的多重變數（若有時）。進一步的說明，請參閱Integer BASIC或Applesoft的手册。

從隨機存取檔讀出

從隨機存取主文檔特定資料錄中進行讀取的過程是略有不同的（請參閱本附錄中的“寫入隨機存取檔”）。隨機存取主文檔中，每一筆資料錄是由檔案開始時指定的L（Length）參數所指定的拜長度所組成，此L參數與當初在檔案寫入之前的OPEN命令中所指定的L參數相同，要計算某一資料錄的開始（READ命令中的R參數所指定）位置，DOS利用L參數來計算前面所有資料錄所佔的拜數。此數除以256（$1,00）即爲DOS要來到含有該資料錄的段落上所須跳過的段落數。然後DOS檢查該檔的T／S表以找到所要段落在磁碟上的位置。最後，DOS將正確的段落讀進檔案緩衝器，以便從中讀取正確的拜。

雖然主文檔在原來存入時是循序檔，或長度（由L參數指定）

完全不同的隨機存取檔，但其取出的過程仍然相同。DOS不管原來在寫入時所用的L參數，只知道利用目前在讀取之前開啓（OPEN）檔案時所指定的L參數來計算所要求資料錄的磁碟位置。

在READ命令中使用R（Record）及B（Byte)兩個參數，可以使下一個INPUT 指令從所指定的資料錄中特定拜位開始讀取。此拜位可以是在目前資料錄中的位置之前或之後。要想有效的使用這個參數，必須對特定資料錄中每一個拜的內容都很清楚才可。

POSITION 命令原來主要是供存取循序檔之用，在此也能利用R（相對欄位）參數中所指定的欄位數以使DOS目前位置指標往前移動。在READ命令中使用R（此處表示Record) 參數會將目前位置指標移動到所要資料錄的開始位置。使用POSITION 會取消READ時所定的模式（並不重新設定位置指標），然後再用READ（此時不須參數）時，又會重新設定到READ模式。

每次開啓（OPEN）檔案時，DOS會忘掉目前在檔中的位置，而又從頭開始讀取（除非指定Byte 或／及Record 參數）。

DOS並不會保持關於主文檔結構、格式、資料錄長度或資料欄長度的情報。因此，要想有效的運用隨機存取主文檔，使用者必須自行將這些情報記錄下來，或存放在檔案開始的部位。

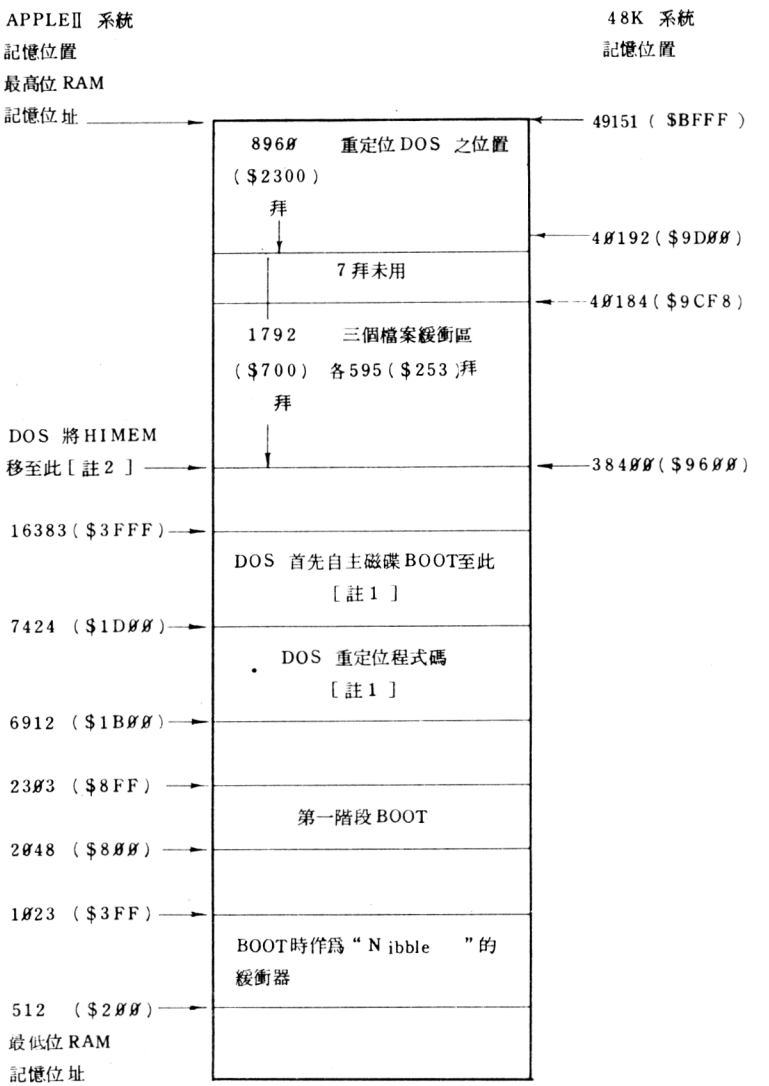
附　錄　D

記憶體之運用

BOOT磁碟時寫入記憶體的區域
DOS及兩種BASIC所使用記憶體的區域
BOOT DOS時所設定的HIMEM值

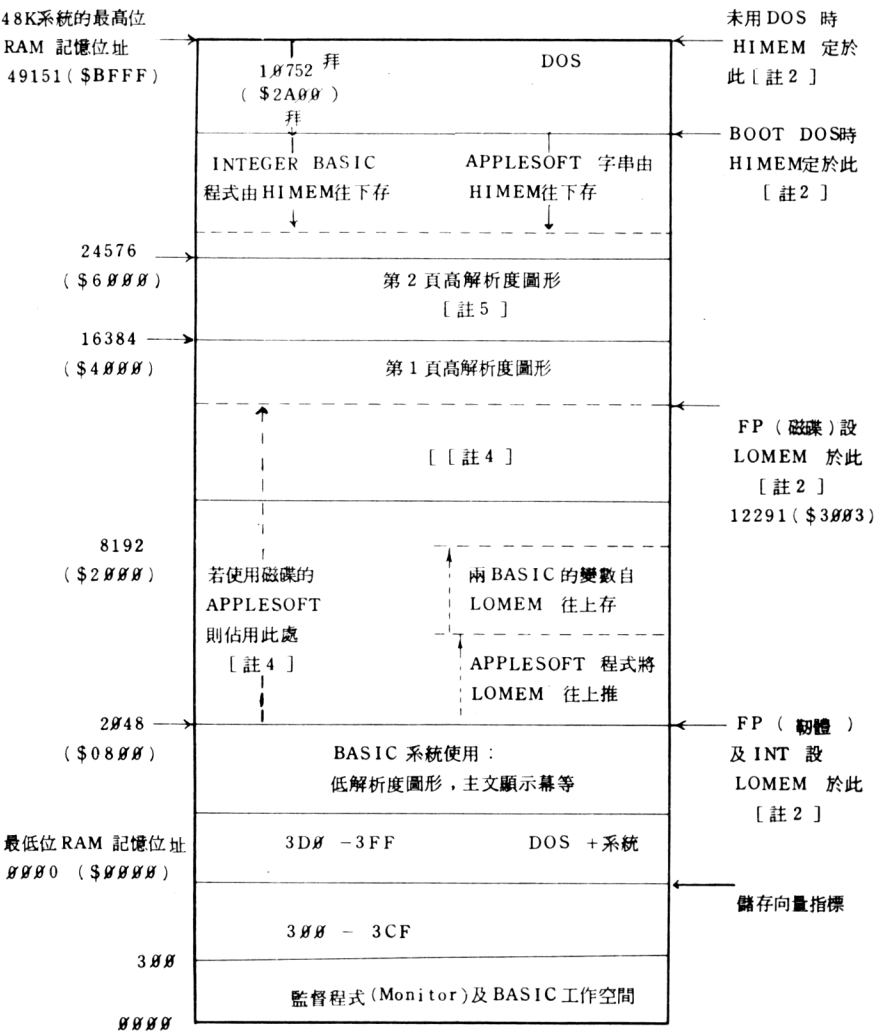
表1：APPLEII 記憶體使用之對映圖

A、**BOOT磁碟時寫入記憶體的區域**



註1：BOOT副磁碟時，本區不受影響。因此時DOS 直接放入(INIT此副磁碟時)系統RAM 之最高區域（此區域的RAM 在此BOOT系統中不一定存在，視本系統記憶容量而定）。

B、**DOS及兩種BASIC所使用記憶體的區域**



註2：若系統處於 INTEGER BASIC ，HIMEM 指標存在76-77 （ $4C-$4C ）位置（依低／高拜順序）。若系統處於APPLESOFT BASIC ，HIMEM指標依相同順序存在115-116 （ $73-$74 ）位置。TABLE2 中可見到由BOOT DOS設的HIMEM指向位址。增加MAXFILES數，使HIMEM下移，每增一個檔案緩衝區，多用了595拜。有關其他APPLESOFT 程式指標的位置，可參考APPLEⅡ BASIC 程式手册之附錄 I 得之。（ Programming Mannual ）

表2：BOOT DOS時所設定的HIMEM值

依系統記憶體大小，BOOT DOS 設定了HIMEM 如下：

系統	最高RAM 位址		HIMEM	
大小	10 進位	16 進位	10 進位	16進位
16K	16383	$3FFF	5632	$1600
20K	20479	$4FFF	9728	$2600
24K	24575	$5FFF	13824	$3600
32K	32767	$7FFF	22016	$5600
36K	36863	$8FFF	26112	$6600
48K	49151	$BFFF	−27136	$9600[註3]

註3：−27136表38400，INTEGER BASIC 接受之最大數目爲32767 。大於32767 之記憶位址以其等效負值表示（即爲n - 65536 ）。

註4：第一頁高解析度圖形的使用，會洗掉8192至16383位置記憶體原有內容。若HIMEM指標低於16383 ，部分DOS亦被洗掉。因此，在記憶體大於32K 系統上才能同時使用DOS及高解析度圖形。

第一頁高解析度圖形與磁碟的APPLESOFT 不能並用。在32K 以上的系統，使用磁碟的APPLESOFT 時，可用第二頁高解析度圖形。

註5：第二頁高解析度圖形佔用16384 至24575 位置的記憶體。若HIMEM指標低於24575 ，部分DOS 會被洗掉。因此，在大於36K 的系統上。才可同時使用DISK II 及第二頁高解析度圖形。

附　錄　E
DOS 進入點與 DISK Ⅱ 界面電路圖

DOS進入點

DISK Ⅱ界面電路圖

DISK Ⅱ類比線路板電路圖

一、DOS進入點

重連接DOS 的常規（第三頁記憶體已寫入情況下）：

系統大小	10 進位位址（CALL）	16 進位位址（G）
48K	-25153	$9DBF
32K	23999	$5DBF
16K	7615	$1DBF

監督程式的3D0L命令可將此值顯示在幕之右上角。

程式經由BLOAD 之後之起始位置及長度資料存放的位置：

系統	起始位置（低拜）		程式長度（低拜）	
大小	10 進位	16 進位	10 進位	16 進位
48K	43634	$AA72	43616	$AA60
32K	27250	$6A72	27232	$6A60
16K	10866	$2A72	10848	$2A60

在BLOAD 之後鍵入下式，可見到程式所在的起始位置及其長度

PRINT PEEK（低拜）+ PEEK（低拜+1）* 256

在任意系統上，下列程式可用來找尋最近一次被BLOAD 程式起始位置及長度資料所存放的位置。

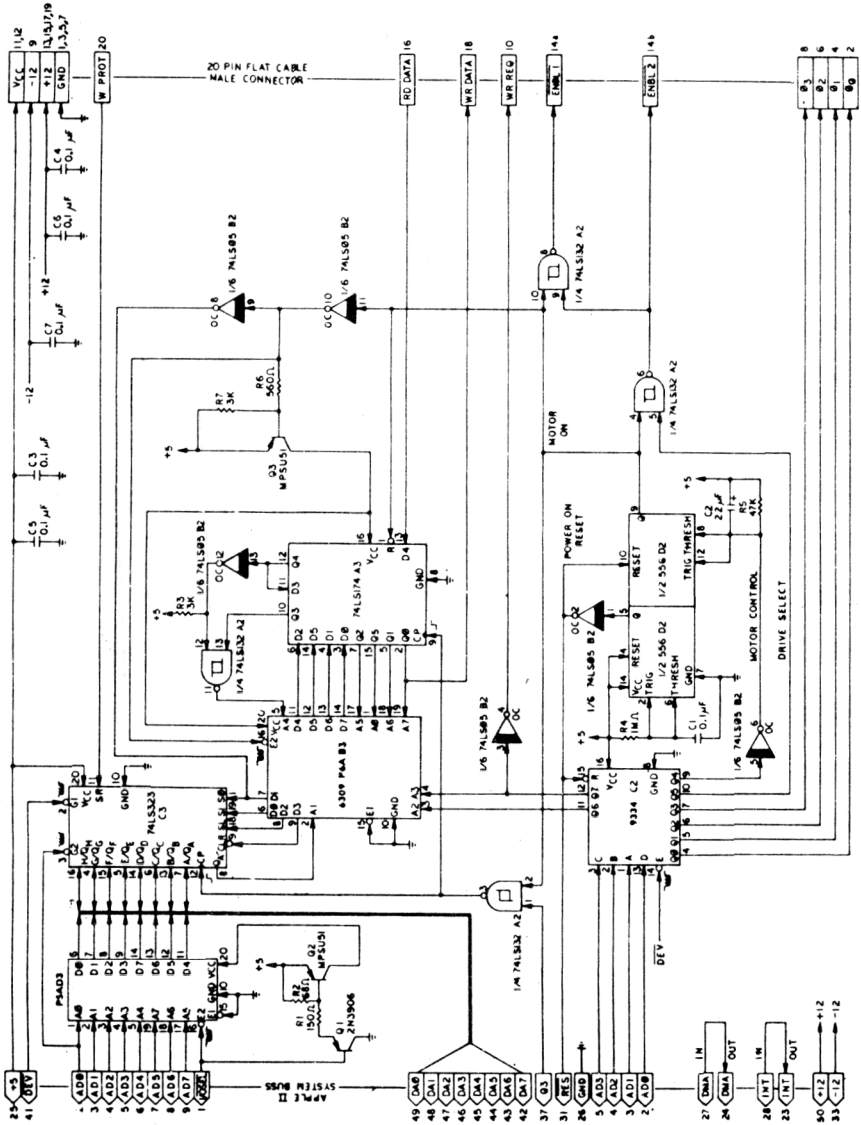
```
5  REM  BLOAD FINDER
7 H = 38400: REM  DOS-BOOT HIMEM
8 T = 49152: REM  HIGHEST ADDRESS
10 D$ =  CHR$ (4): REM CTRL-D
20  PRINT D$; "BSAVE FOO, A$7777, L$77"
30  PRINT D$; "BLOAD FOO"
40  PRINT D$; "DELETE FOO"
50  FOR I = H + 1792 TO T
60  IF  PEEK (I) < > 119 OR  PEEK
     (I + 1) < > 119 THEN  NEXT I
70  PRINT "LOCATIONS OF START
     ADDRESS: "; I; ", "; I + 1
80  FOR I = H + 1792 TO T
90  IF  PEEK (I) < > 119 OR  PEEK
     (I + 1) < > 0 THEN  NEXT I
100  PRINT "LOCATIONS OF LENGTH: "
      ; I; ", "; I + 1
```

其中H及T爲在48K系統上之值。其他系統上應有的值可見附錄D的表2。本程式約耗兩分鐘找出正確位置。

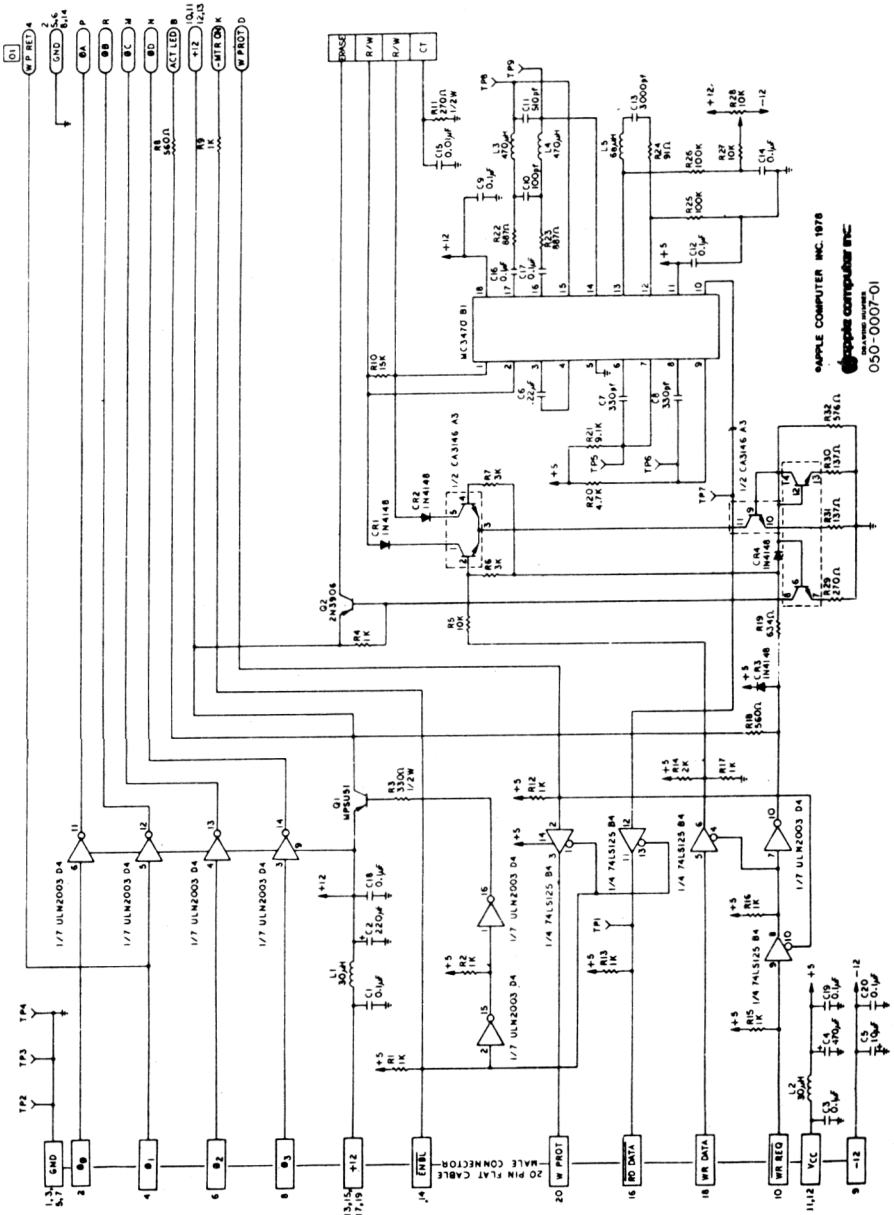
DOS 字元的輸入及輸出常式：

參考第10章（其註7），用此方法的例子可見於附錄F“檔案名稱”部分的程式。

二、DISK Ⅱ界面電路圖



三、DISK Ⅱ類比線路板電路圖



附　錄　F

DOS 命令總整理

符號意義
檔案名稱
管家命令
存取命令
循序主文檔的命令
隨機存取主文檔的命令
機器語言檔的命令

DOS命令可分爲五個種類：

管家命令：

INIT	RENAME	VERIFY
CHALOG	DELETE	MON
SAVE	LOCK	NOMON
LOAD	UNLOCK	MAXFILES
RUN		

存取命令：

FP	PR#	CHAIN
INT	IN #	

循序主文檔的命令：

OPEN	WRITE	POSITION
CLOSE	APPEND	EXEC
READ		

隨機存取主文檔的命令：

OPEN	READ
CLOSE	WRITE

機器語言檔的命令：

BSAVE

BLOAD

BRUN

一、符號意義(Notation)

命令結構的語法。以簡單的符號表示DOS命令的語法。

DOS命令之後的中括號〔〕內含命令所須的參數項目。這些參數項目依須要而被採用。除了特殊規定外這些參數項目的順序可任意安排。

在命令含檔案名稱情況下，檔案名稱須緊跟於命令文字後(空一格)，在檔案名稱與參數間以逗點分之。

{CTRL}　表示按着CTRL鍵並按其他鍵。如{CTRL}D表按着CTRL鍵並鍵入D。CTRL—D表同樣意思。

{RETURN}　表示按下RETURN 鍵。

{RESET}　表示按下RESET 鍵。

{ESC}　表示按下ESC鍵。

各命令表示法中，大寫字母及逗點應照規定鍵入，而其中小寫字母代以所要填入的參數。小寫字母的意義如下：

f：檔案名稱。可爲1至 3Ø 個字元組成。除逗點以外的任何字元皆可在內，唯第一字元限爲英文字母。

例如：CHESS

SUM OF SQUARES

NEW 45

HOW-ABOUT-THIS

g：第二個檔案名稱。

s：溝槽號數。s表APPLEⅡ 之磁碟控制卡所在的溝槽位置(一般爲第六槽)。s的初設自然值(default)設在BOOT DOS時磁碟控制卡所在溝槽。變更或使用過s值後，每次所用的s自然值定於前一次取用時之值。s值範

爲1－7。

若磁碟控制卡不存在所指定的s溝槽上，系統可能停止工作，記憶體中之程式亦可能失去。參看附錄B的I／O錯誤誤部分。

v：磁碟册號數。v之初設自然值定爲所BOOT磁碟的册號數。在其後使用到v之自然值皆同於其前一次所用之v值（在CATALOG命令，v可能受其內部設定）。v值範圍爲 0－254。

註：DOS命令中，定義册號數爲0或不設v值時表示令DOS採用並定v值同於所使用磁碟的册號數。

d：磁碟機號數（1或2）。d之初定自然值爲1。其後使用的d自然值同於前一次所用的d值。

p：POSITION與EXEC命令在對循序主文檔作用時，須用到R與P參數。p指向現在檔案中位置往後數的第p欄。p之自然值定爲0，p爲零不移動檔案中的"檔案位置指標"。

註：EXEC命令常將指標設在檔案起位置，故用EXEC命令時p常設爲0。p的範圍爲0－3276。

r：資料錄號數。READ及WRITE命令在對隨機存取主文檔作用時，須用到R參數。OPEN檔案之後，其r自然值定爲0，而在之後使用中其自然值皆定於前一次使用之r值。隨機存取檔中，r指向資料錄的絕對位置（檔中第r資料錄）。r的範圍爲0－32767。

a：表RAM中某位址。BSAVE命令須用此參數。a爲BSAVE及BLOAD資料在記憶體中之起始位置。若

BLOAD未指定a值，則載入當初BSAVE 時a所指的起始始位置。a之範圍爲Ø－65535 。

b：拜數，b之自然值定爲Ø。用在循序檔時，b指向檔案的一個絕對位置拜上。用在隨機存取檔時，b指向r指定資料錄中的一個絕對位置拜上。b之範圍爲Ø－32767 。多數應用上，b之範圍在Ø至序向檔之前一次使用所指之拜或隨機存取檔指定資料錄上之前一次使用所指之拜。

j：指定長度。j之自然值定爲1 。OPEN命令對隨機存取檔作用時用到此參數。j決定隨機存取檔一個資料錄所擁有的拜長度（拜數量）。BSAVE 命令中，j決定存入磁碟的記憶體內容之拜數量。j之範圍爲Ø－32767 。

應用範例如下：

INIT f [,Vv] [,Ss] [,Dd] 之應用可爲

INIT HELLO , V17 , D2

其中大寫字母及逗點照所示鍵入。“ f ”代以檔案名稱“ V17 ”表示設磁碟爲第 17 册，“ S s ”本例中不用，“ D 2 ”表使用於第2號磁碟機。

DOS命令中之常數亦可以 16 進制表示，如$11表17。

二、檔案名稱

檔案名稱可長達 3Ø 個字元，其中不可含逗點或結束命令（CTRT-M ,RETURN）。第一個非空白字元前的空格不算，超過30個字元的字忽略不計。

鍵入檔案名稱時，若用到ESC，左／右箭號，CTRL－C，CTRL－H等特殊鍵會造成不可預期的結果。

檔案名稱中若含其他控制符號，在一般情形下此控制符號不顯示出來，但使用或除掉此檔時，此控制符號仍須鍵入。

以下程式可找出除了CTRL—M（RETURN），ESC，CTRL—H（左箭號）及CTRL—U（右箭號）之外的任何隱藏的控制符號。

```
10  DATA 201,141,240,21,201,136
20  DATA 240,17,201,128,144,13
30  DATA 201,160,176,9,72,132
40  DATA 53,56,233,64,76,249
50  DATA 253,76,240,253
60  FOR I = 768 TO 768 + 27
70  READ V: POKE I,V: NEXT I
80  POKE 54,0: POKE 55,3
90  CALL 1002
```

若不確定檔案名稱中是否有控制符號，則可鍵入此程式，SAVE並RUN之。等]出現，再用CATALOG命令便可得到含閃動控制符號的檔案列表。若欲回復正常的印出方式，可鍵入PR#0。

三、管家命令

INIT f [,Vv] [,Ss] [,Dd]

例如：INIT HELLO , V18

V18表示設所INIT之磁碟爲第 18 册磁碟。第2章及附錄G中中詳述INIT磁碟。

CATALOG [,Ss] [,Dd]

例如：CATALOG

在顯示幕上印出被指定（或自然值定）的磁碟機上所用磁碟的

册號數及其內含的所有檔案列表。而此後，册號數的自然值被定爲此磁碟的册數號。若本命令中含册號數參數[,Vv]，本命令執行時忽略之。

CATALOG將每個檔案的檔案型式及其所佔磁碟段數一起顯示出。檔案形式如下：

I：表以SAVE建的INTEGER BASIC 程式檔。

A：表以SAVE鍵的APPLESOFT BASIC 程式檔。

T：表以OPEN建立並以WRITE 填入內容的主文檔（ Text file ）。

B：表以BSAVE 鍵的二進位的記憶體映像檔（ Binary memory-imag file ）。

在檔案形式符號旁若含星號表示此檔案已被鎖制（ Locked ）

磁碟中使用者至多可用到 496 個段落，每個段落存 256 拜資料。檔案至少佔一個段落。例如空的主文檔即佔一個段落（此檔案的空"T／S表"佔用一段落）。空的INTEGER BASIC,APPLESOFT,及機器語言檔須佔 2 個磁碟段落（一個爲其"T／S表"，另一爲第一程式段落，存程式長度——參看附錄C）。

檔案超過 255 段落時，CATALOG之顯示長度扣除 256 之後自 000 開始表示出來。此舉對檔案不影響，只對磁碟內佔用情形造成混淆不了解。

SAVE f [,Ss] [,Dd] [,Vv]

例如：SAVE COLOR DEMOS , V56

在指定或自然值定的磁碟機上之磁碟建立一以所給名稱爲名的檔案，並將 INTEGER BASIC 或APPLESOFT 程式存入檔中。

若磁碟上原已有此同名檔案，卻屬不同語言或檔案型式，則顯示“FILE TYPE MISMATCH ”但若其具相同語言型式，則將此原檔案內容代以現有的BASIC 程式，且不對使用者作任何警告。

LOAD f [,Ss] [,Dd] [,Vv]
例如：LOAD DOW JONES , V19 , D1

在指定或自然值定的磁碟機上之磁碟中找出名爲 f 的 INTEGER BASIC 或APPLESOFT 程式。若磁碟册號數符合，且存在此檔案，則將程式載入記憶體，之後此程式便可被LIST、RUN或SAVE。LOAD命令會關閉（ CLOSE ）任一“開主文檔”,會將APPLE 轉入與 f 檔相同的語言模式，並會消除記憶體中之原有程式。

當檔案 f 爲APPLESOFT BASIC 程式，且APPLESOFT未在記憶體，APPLESOFT 之ＲＯＭ卡亦未使用，則APPLE-SOFT會自動由指定磁碟機LOAD入並ＲＵＮ之。若APPLESOFT既不在磁碟又不在ＲＯＭ卡上，則會顯示出

LANGUAGE NOT AVAILABLE

不含參數的LOAD命令亦可用來自磁帶載入程式。

RUN f [,Ss] [,Dd] [,Vv]
例如：RUN ANNUITY , D2

自指定或自然值定的磁碟上之磁碟LOAD入檔案 f ，並ＲＵＮ之。ＲＵＮ已在記憶體中的程式，只要鍵入ＲＵＮ即可。

RENAME f,g [,Ss] [,Dd] [,Vv]
例如：RENAME SEPERATE , SEPARATES4,D1,V0

在指定或自然值定的磁碟機上之磁碟找出f檔，並將其名改爲g，檔案內容不變。若f已OPEN則CLOSE它。

RENAME 命令不檢查g是否重覆使用，因此可以利用RENAME 將許多檔案設以相同名稱——此舉造成潛在的混淆。

DELETE f [,Ss] [,Dd] [,Vv]
例如：DELETE TEST

在指定或自然值定的磁碟機上之磁碟除掉名爲f的檔案。若f已OPEN，此命令CLOSE之。參看附錄C。

若f不存在磁碟，則顯示出

FILE NOT FOUND

爲了避免因此停掉你的程式，可先OPEN此檔案再DELETE它。

LOCK f [,Ss][,Dd][,Vv]
例如：LOCK LOVE LETTERS , V31

保護存在指定或自然值定的磁碟機上之磁碟中f檔案免於被更改或除掉。CATALOG命令上，會以星號標出已LOCK的檔案。

UNLOCK f [,Ss][,Dd][,Vv]
例如：UNLOCK RECIPES , V31 , D2

若要更改或除掉已鎖制的f檔，則先利用此命令解鎖之。

VERIFY f [,Ss][,Dd][,Vv]
例如：VERIFY SAM

檢驗存入磁碟的 f 檔之資料是否完整。（在檔案以 SAVE，BSAVE 或WRITE 建立時，DOS計算輸出緩衝區的內容得到查核碼，其後將查核碼同緩衝區內容一起存入磁碟段落。VERIFY 命令自每個檔案段落的內容計算出一新的查核碼，再與前述存入的查核碼比較）。若檢驗結果無誤，則不發出任何信號。但若檢驗結果，則顯示

I／O　ERROR

任何型式檔案皆可以VERIFY 查驗之。

MON [C][,I][,O]
例如：MON O
　　MON C , I , O

平常在計算機與磁碟間的資料或命令傳送並不顯示在幕上。MON命令爲一有力偵錯工具，它可令傳送之部份或全部資料顯示於幕上。選C，磁碟命令會顯示出。選I，自磁碟輸入至APPLE-Ⅱ的資料會顯示出。選出O，自APPLE Ⅱ輸出至磁碟的資料會顯示出。C，I，O三者至少選一個，否則MON命令不作用。三參數次序不限而以逗點隔之。

註：MON功能直到出現NOMON、改用語言（FP或INT）、或重置（3D0G）才停止作用。

NOMON [C][,I][,O]
例如：NOMON I,C

NOMON 可終止MON的作用（部分或全部作用）。NOMON C，I，O將系統設回原來正常狀態。選C，磁碟命令不顯示出。選I，自磁碟輸入APPLE 的資料不顯示出來。選O，自APPLE輸出至磁碟的資料不顯示出來。C，I，O三者至少選一個，否則NOMON 命令不作用。三參數次序不限而以逗點隔之。

MAXFILESn

例如：MAXFILES 6

其中n爲1至 16 的整數。n定出可同時工作的檔案數目。MAXFILES保留 595 拜的記憶體作爲一檔案緩衝區供一個檔案使用。BOOT系統時，n之自然值爲 3，故保留 1785 拜作爲三個檔案緩衝區，同時可OPEN3 個檔案。除PR#，IN#，及MAXFILES外之其他DOS命令的執行即要用到檔案緩衝區。若所訂的所有檔案緩衝區都用完，再執行DOS命令時（如：CATALOG）會顯示下列信號。

NO BUFFERS AVAILABLE

MAXFILES 命令會使HIMEN 移動，而可能破壞掉INTEGER BASIC 程式或APPLESOFT 程式的串敍述（STRING）。MAXFILES 應用在LOAD及RUN之前。若欲將MAXFILES命令用在程式中，參考第五章及第七章。

四、存取命令

FP [,Ss][,Dd][,Vv]

例如：FP，D2

本命令將系統設成APPLESOFT BASIC 。記憶體中原有的INTEGER BASIC或APPLESOFT 程式皆失掉。若系統原含有APPLESOFT ROM卡，DOS不顧卡片上開關位置直接使用其上的語言。若系統不含APPLESOFT ROM 卡，DOS便會LOAD及RUN磁碟中的APPLESOFT 程式（於指定或自然值定的磁碟機上之磁碟）。

欲將APPLESOFT 程式放入新INIT的磁碟，須先將它LOAD入APPLE 中再SAVE APPLESOFT至新INIT的磁碟。

勿以RUN APPLESOFT改變語言模式。此舉，表面上可能沒問題，但不保險。爲避結果可能一團糟，最好使用FP 。

INT

例如：INT

將系統設爲INTEGER BASIC。執行此命令後，記憶體中任何何INTEGER BASIC或APPLESOFT 程式皆失掉。

CTRL-D （{CTRL}D）

例如：10 D$=CHR$(4)

20 PRINT D$;"WRITE CHESS"

APPLE 上，每個以PRINT 輸出的字元都經由DOS檢查過，若PRINT 一" RETURN "字元（ PRINT 敍述常以一"RETURN"控制字元結束之），並跟隨一CTRL－D，則表其後的一串字元爲DOS命令（至下一個" RETURN "字元爲止）。DOS的大多數命令可以藉此方法用在INTEGER BASIC或APPLESOFT程式中。

用法爲，先建一僅含CTRL-D 的字串D$，然後依上例做之。在APPLESOFT 中可利用CHAR$(4)建D$。還有一種方法是將CTRL-D 鍵入兩引號間，但此時CTRL-D 並不顯示出，詳細情形，參考附錄G。

PR# s
例如：PR#6

將此後的APPLE 輸出自TV顯示幕改至以第六槽控制卡控制的輸出裝置。PR# ø 表示控制其回至TV顯示幕輸出。此命令若用於程式中，寫成：

```
1ø D$ = "" : REM CTRL-D
2ø PRINT D$ ; "IN# 1"
```

若在s槽上並無輸出裝置控制卡，系統會停止工作，使用者須以RESET 鍵回復之。

IN# s
例如：IN# 6

將此後APPLE 的輸入自鍵盤改爲由第六槽路控制卡控制的輸入裝置。IN# ø 控制其回至鍵盤輸入。此命令若用於程式中，寫成：

```
1ø D$ = " " : REM CTRC-D
2ø PRINT D$ ; " IN# 1 "
```

若在s槽上並無輸入裝置控制卡，系統會停止工作，使用者須以RESET 鍵回復之。

CHAIN f [,Ss][,Dd][,Vv]
例如：CHAIN PART TWO,D1,S7,V0

用於INTEGER BASIC程式中,它將指定或自然值定的磁碟機上磁碟的f檔（INTEGER BASIC程式）LOAD並RUN之，不清除任何變數值。如此一來，f便可根據前一次執行過的程式結果運算之並留下運算結果供後來程式使用。APPLESOFT 程式不能利用此命令。在這方面可參考第10章或附錄G中所述APPLE-SOFT程式的特殊程序。

五、循序主文檔的命令

OPEN f [,Ss][,Dd][,Vv]
例如：OPEN SESAME , D2

本命令配置一個595拜的記憶體緩衝區供f檔案使用，並令系統準備自檔案起始點寫入或讀出資料。本命令與WRITE 或READ連用，以建立（Create）或取回（Retrieve）循序主文檔。

若指定或自然值定的磁碟機上之磁碟並無f檔，則建立此檔（Create）。若f檔已OPEN，本命令先CLOSE 之，再OPEN所需要的檔案。

CLOSE [f]
例如：CLOSE WINDOV

若原來正在WRITE 資料，CLOSE 命令會將檔案緩衝區的輸出部份所有剩餘字元寫入指定磁碟中已OPEN的檔案。CLOSE f取消掉已配置給循序主文檔f的檔案緩衝區。CLOSE 命令若未指定檔案名稱，則除了EXEC檔外所有已OPEN的檔案都CLOSE 掉

。（在一時間只能OPEN一個EXEC檔，欲OPEN另一EXEC檔，已有的EXEC檔便自動CLOSE 掉）。

當主文檔OI　時，程式當若被CTRL - C中斷，則最好鍵入CLOSE 以免遺失掉已有的資料。

檔案以OPEN被配置緩衝區後，須以CLOSE 結束之，否則可能能造成資料遺失。

WRITE f [,Bb]

例如：WRITE ADDRES DATA

本命令作用之後出現的PRINT 敍述將其輸出送到指定的檔案而不送到APPLE 的TV顯示幕。b參數表自檔案第b拜開始寫入（參考第六章）。WRITE 的功能在遇到任意一DOS命令或INPUT 敍述之後取消掉。故可利用一個的DOS命令（如：PRINT 一個CTRL - D）以控制其停止作用。WRITE 命令使用於延遲執行模式（defer-execution mode）。

本命令作用後所有的APPLE 輸出字元（包括INPUT 的催促信號“？”，錯誤訊息及其他未預期的字元，皆輸至磁碟而不送至顯示幕。

READ f [,Bb]

例如：RESAME

本命令作用之後出現的INPUT 敍述（APPLESOFT中GET敍述）自所指定的循序主文檔f取得對應字元，而不從鍵盤取之。b參數表示自檔案第b拜開始讀取。

每一 INPUT 敍述只能自循序主文檔讀入一欄（ Field ）的資料。一欄的長度可能自 1 至 32727 個字元，而以“ RETURN ”字元分之。因字串及 I／O緩衝區容量的限制，故不易一次存取一個大於 255 個字元的欄 。

PRINT 任一DOS命令可取消READ功能。可利用一無作用的DOS命令（如： PRINT 一個CTRD - D ）控制其停止作用。READ命令使用於延遲執行模式。

APPEND f [,Ss][,Dd][,Vv]
例如：APPEND MORE INFO

本命令OPEN所指定的主文檔，並將檔案之定位指標（ position-in-the-file pointer ）指到檔案最末處。APPEND 命令之後跟隨一具有相同 f 的WRITE 命令。APPLEND 之後的下一寫入字元跟隨在之前依順序進入檔案的字元後面。（ APPEND 之後不可跟一OPEN命令，因OPEN會將檔案之定位指標移到檔案開端處。

POSITION f [Rp]
例如：POSITION ADDRESS.DATA , R277

POSITION 爲一相對，非絕對位置的定位命令。它將檔案的定位指標移至原所在位置往下數第 p 欄資料的起始點。資料欄爲一以“ RETURN” 字元爲終止記號的一串字元。本命令執行後READ或WRITE 自此新定位置開始讀或寫資料 。

POSITION命令之作法是，自原位置往後檢查檔案中每一字元，直到找出第 p 個“ RETURN” 字元，之後將檔案定位指標設在此“ RETURN” 字元之後一拜。檢查過程中若發現任一拜中未存字

元，則打出以下訊息：

END OF DATA

則可能第p欄已超過檔案內容的最後位置。

EXEC f [,Rp][,Ss][,Dd][,Vv]
例如：EXEC UTILITY

本命令與RUN功能類似，但其f爲一內含BASIC及DOS命令的主文（資料）檔。當執行時這些DOS命令如同由鍵盤進入作用之。利用本命令，使用者可以建一控制APPLE的檔案，使其動作同於由鍵盤上控制一樣。

同一時間中只能使用一個EXEC命令，若在EXEC檔中出現另一EXEC命令時，原來的EXEC檔則被CLOSE掉，然後OPEN並執行新的EXEC檔。EXEC命令OPEN f 檔並執行時，若其中含有CLOSE命令，正進行EXEC的f檔不被CLOSE掉。當EXEC檔做完所有敍述後，它才將自己CLOSE掉並停止動作。EXEC檔中若含RUN程式，EXEC會等RUN執行完再跳至下一命令上繼續做。

當一EXEC檔已OPEN而RUN一程式時，其任何INPUT敍述都從EXEC檔的次一欄讀入對應資料而不自鍵盤讀入。若此欄對應資料竟是一DOS命令，則先執行此命令，之後再繼續做程式。

當EXEC檔仍OPEN，而以CTRL - C終止正在RUN的APPLESOFT程式，則EXEC檔中“RUN”以後的剩下未執行指令亦不再執行。

若設定p參數，檔案的檔案定位指標便指向第p欄的起始點，而EXEC便自此處開始執行。p參數亦如在POSITION命令中同爲相對位置的參數，但EXEC自檔案開端數起，故p的參考點爲0。

六、隨機存取主文檔的命令

OPEN f , Lj [,Ss][,Dd][,Vv]
例如：OPEN SEAME , L2

本命令爲隨機存取主文檔f配置一個595拜檔案緩衝區，並設其資料錄（Record）長度爲j拜。j值範圍爲1－32767，其自然值定爲（default）1。

OPEN與READ或WRITE連用，以建立（Create）或取出（Retrieve）隨機存取主文檔。隨機存取主文檔應定義其資料錄長度，因此L參數必須定義。每當再度用到此檔，須以相同的L參數OPEN之。DOS利用此值計算出所指定資料錄的起始點。

若磁碟中無f檔，本命令自動建一f檔。

CLOSE [f]
例如：CLOSE BOOK

若使用者正WRITE檔案，本命令會先將檔案緩衝區輸出部份所餘的字元送至磁碟，再CLOSE此檔案。CLOSE除掉了配置給隨機存取主文檔f的檔案緩衝區。若CLOSE命令上不指定檔案名稱，則除了EXEC檔外的所有已OPEN的其他檔案都會被CLOSE掉。若在主文檔OPEN時，程式被CTRL－C中斷，則最好鍵入CLOSE以免遺失掉已有的資料。

已OPEN的檔案須適時CLOSE掉，否則可能遺失資料。

WRITE f [,Rr][,Bb]
例如：WRITE ADDRESS.DATA , R3

WRITE 命令之後出現的 INPUT 敍述會將其資料輸至所指定的檔案，而不送至顯示幕。WRITE 的功能直到 PRINT 任一 DOS 命令，或一 INPUT 敍述才結束。可以用一個無作用的DOS命令（如：PRINT 一個CTRL - D）簡單地結束其功能。WRITE 只能用在延遲執行的 PRINT 敍述。

R參數使WRITE 命令自檔案的第 r 資料錄寫入。其資料長度爲OPEN檔案時所定的 j 值。r 的自然值定爲 0。B參數使WRITE命令自檔案第 r 資料錄的第 b 拜寫入。

WRITE 之後的所有APPLE 輸出字元包括 INPUT 問號“？”，錯誤訊息及其他非預期的字元都送至磁碟檔案而不送至顯示幕。

READ f [,Rv][,Bb]
例如：READ SESAME , R3 , B30

READ命令之後出現的 INPUT 敍述（APPLESOFT 中的GET敍述）自所指定的隨機存取主文檔讀取對應的字元而不自鍵盤取之。每一個 INPUT 敍述自檔案中所指定資料錄上讀入一欄資料。每一欄可具 1 至 32767 個字元而以RETURN 結束之（不可超過 j 長度限制）。

R參數使READ命令自第 r 資料的第一拜讀入。r 的自然值爲 0。B參數使READ命令自第 r 資料錄的第 b 拜開始讀出。

READ的功能以 PRINT 任一DOS命令而取消之。故可利用一個無作用的DOS命令（如PRINT 一個 CTRL - D）簡單地取

消之。

七、機器語言檔的命令

BSAVE f , Aa , Lj [,Ss][,Dd][,Vv]

例如：BSAVE PITURE , A16384 , L8192

BSAVE PICTURE , A$4000 , L$2000

本命令建立一檔案以儲存APPLE Ⅱ記憶體中某段（Segment）的內容。起始位置a及所欲存之拜數量j兩參數定義了此段位置。

上例將高解析度影像（第二頁）存入磁碟中機器語言檔f。兩式式作用相同，第二式之數值以十六進位制表示出。

BLOAD f [,Aa][,Ss][,Dd][,Vv]

例如：BLOAD PICTURE , A8192

BLOAD PICTURE , A$2000

a參數令BLOAD將資料自APPLE Ⅱ記憶位址a開始載入。若命令中未含a參數則BLOAD命令將所指定的檔案內容放回它在被BSAVE之前原來所處的記憶位置。（注意：有些機器語言程式在移動其位置之後會變得無法正常執行）。

在非使用ROM卡的APPLESOFT時（用磁碟的APPLESOFT時）上例不可使用（因記憶體佔用問題）。

BRUN f [,Aa][,Ss][,Dd][,Vv]

例如：BRUN SUPER , A$C00A , V75

本命令首先將f檔BLOAD入記憶體（自a位址開始，若未給

給定A參數，則放回原BSAVE時所在位置），之後程式跳越至a處執行f檔的機器語言程式。

附　錄　G

DOS 使用程序

BOOT DOS

INIT磁碟

自意外的RESET返回原狀態

在程式中利用DOS

建—TURNKEY系統

建立及取出循序主文檔

將資料加入循序主文檔

經由循序主文檔控制APPLE

建立及取出隨機存取主文檔

複製主文檔

APPLE SOFT的CHAIN功能

本附錄說明DOS 中許多重要的工作程序。

一、BOOT DOS

催促符號	使用的語言	應鍵入
>	INTEGER BASIC	PR#s 或 IN#s
]	APPLESOFT	PR#s 或 IN#s
*	監督程式	s {CTRL}P

註：其中s表槽路號數

二、INIT 磁碟

INIT 副磁碟方法爲：

1. BOOT DOS
2. 將空磁碟插入磁碟機。
3. 鍵入GREETING 程式。例如：

```
10 PRINT "32K SLAVE DISKETTE INIT-
    IALIZED 5 MAY 80"
20 END
```

4. 若欲令GREETING 程式名爲HELLO，則：INIT HELLO
5. 在磁碟機"IN USE"燈熄滅後，取出磁碟並標示其册號。

若欲複製主磁碟(Master disk)，必須使用Master Create 程式。

三、自意外的RESET返回原狀態

已BOOT DOS 後，若無意中按了RESET鍵，可以鍵入"3D0G"以返回原使用的BASIC程式。

四、在程式中利用DOS

PRINT一個CTRL-D 可以將DOS 命令用在使用者程式之中。在APPLESOFT上，可先設D$ 使其內容爲CTRL-D 字元。

依下 式得D$ ：

D$=CHR$(4)　　（CTRL—D 的ASCII 碼爲4 ）

另一種方法INTEGER BASIC及APPLESOFT皆可用，就是爲鍵入D$=''再鍵入CTRL—D ，之後再鍵入''，在顯示幕上CTRL—D 不出現，只顯示D$=""。

RUN 下列APPLESOFT程式會做CATALOG 命令

```
5  REM.  GREETING PROGRAM
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"CATALOG"
```

一個PRINT敍述只能用一個DOS 命令，PRINT敍述以CTRL—D 開始，以DOS 命令結束。CTRL—D 前有一RETURN字元（PRINT 敍述結束後自動送出RETURN 字元）。

PRINT敍述的CTRL—D 之後的下列命令，只用於延遲執行模式（程式中）。

OPEN,APPEND,READ,WRITE,POSITION

而INIT及MAXFILES最好用於立即執行模式（不用在程式中）。其他的DOS 命令可用於任一模式。

五、建一TURNKEY系統

建一磁碟使其每當BOOT，便RUN 一特定程式。以COLOR DEMO爲例，方法如下：

1. INIT一個空磁碟，以HELLO爲GREETING 程式名稱。
2. 插入含COLOR DEMO 程式的磁碟，鍵入RUN COLOR DEMO 確定執行無誤後，返回BASIC。
3. 插入新磁碟，並鍵入SAVE HELLO 以取代原有的HELLO 程式。

六、建立(Create)及取出(Retrieve)循序主文檔

先用OPEN 再用WRITE命令建立循序主文檔，當執行WRITE 命令後，所出現的PRINT敍述將其資料字元送至磁碟檔上。寫完之後CLOSE此檔。WRITE的功能以PRINT任一DOS命令或遇到一INPUT敍述而結束(PRINT 一個CTRL—D 即可)。

以下APPLESOFT程式建立一個名爲SAMPLE 的循序主文檔，此檔含13欄，包括三字元串及整數1 到10 。

```
5  REM  MAKE SAMPLE
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN SAMPLE"
30  PRINT D$;"DELETE SAMPLE"
40  PRINT D$;"OPEN SAMPLE"
50  PRINT D$;"WRITE SAMPLE"
60  PRINT "HI HO": PRINT "HI HO"
70  PRINT "OFF TO THE DISK WE GO"
80  FOR J = 1 TO 10
90  PRINT J: NEXT J
110  PRINT D$;"CLOSE SAMPLE"
```

OPEN一已存在的檔案，並WRITE它，可能會對原檔案內容重覆寫入，致更改原檔內容。

以下APPLESOFT程式取出SAMPLE 檔。可鍵入MON I觀察其內容，MON I將來自磁碟的輸入資料顯示於幕上

```
5  REM  RETRIEVE SAMPLE
10 D$ =  CHR$ (4): REM   CHR$(4)
     IS CTRL-D
20  PRINT D$;"OPEN SAMPLE"
30  PRINT D$;"READ SAMPLE"
40  INPUT A$,B$,C$
50  FOR I = 1 TO 10
60  INPUT W
70  NEXT I
80  PRINT D$;"CLOSE SAMPLE"
```

先用OPEN　再用 READ　命令以取出檔案，當執行READ命令後，所出現的任一 INPUT敍述（APPLESOFT 中的GET亦同）自磁碟檔取出對應資料字元，而不自鍵盤取。輸入完畢後，須CLOSE檔案。READ　的功能以PRINT任一DOS　命令結束之。

七、將資料加入循序主文檔

以下APPLESOFT程式將“TEST　1”　及“AND NOW FOR　TEST”兩資料串加入循序主文檔 SAMPLE 中。此兩字串進入檔案多加的兩欄中。

```
5  REM   APPEND SAMPLE
10 D$ =  CHR$ (4): REM   CTRL-D
20  PRINT D$;"APPEND SAMPLE"
30  PRINT D$;"WRITE SAMPLE"
40  PRINT "TEST 1"
50  PRINT "AND NOW FOR TEST 2"
60  PRINT D$;"CLOSE SAMPLE"
```

八、經由循序主文檔控制APPLE

RUN 以下APPLESOFT程式會建立一名爲DOIT　之主文檔，內含數個DOS　命令：

LIST　20,50

RUN　HELLO

CATALOG

```
5  REM   MAKE DOIT
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN DOIT"
30  PRINT D$;"WRITE DOIT"
40  PRINT "LIST 20,50"
50  PRINT "RUN HELLO"
60  PRINT "CATALOG"
70  PRINT D$;"CLOSE DOIT"
```

建好DOIT 之後EXEC DOIT,DOIT中的三個DOS 命令便一一執行，如同在鍵盤上控制一樣。

九、建立及取出隨機存取主文檔

下列APPLESOFT程式建立一名爲RA—FILE 的隨機存取主文檔，其資料長度爲30 拜。並寫入“NAME ADDRESS”及所對應的資料錄號數於檔案的第12 至15 資料錄中。程式的第70 及80 行中之第13 資料錄內容在最後改爲“DOS”。

```
5  REM  MAKE RA-FILE
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN RA-FILE"
30  PRINT D$;"DELETE RA-FILE"
40  PRINT D$;"OPEN RA-FILE,L30"
50  FOR I = 12 TO 15
60  PRINT D$;"WRITE RA-FILE,R";I
70  PRINT "NAME ADDRESS ";I
80  NEXT I
90  PRINT D$;"WRITE RA-FILE, R13"

100  PRINT "DOS"
110  PRINT D$;"CLOSE RA-FILE"
```

下列APPLESOFT程式READ 隨機存取主文檔RA—FILE的第12 至15 資料錄。讀入之前須先設定資料錄號數（程式中第40 行）。程式之第60 行檢查了每一資料錄輸入字元串A$ 的前3個字元。若此三字元爲“DOS”則打出“RECORD r WAS CHANGED”訊息，再繼續執行。

```
5  REM  RETRIEVE FA-FILE
10 D$ =  CHR$ (4): REM  CTRL-D
20  PRINT D$;"OPEN RA-FILE, L30"
30  FOR J = 12 TO 15
40  PRINT D$;"READ RA-FILE, R";J
50  INPUT A$
60  IF  LEFT$ (A$,3) = "DOS" THEN
      PRINT "RECORD ";J;"WAS CHAN
     GED."
70  NEXT J
80  PRINT D$;"CLOSE RA-FILE"
```

十、複製主文檔

以LOAD 或BLOAD BASIC 或BINARY 檔案內容進入APPLE，再SAVE 或BSAVE此內容到另一磁碟，則完成了複製檔案。採用BSAVE命令時，須設定起始位置及長度參數。

COPY程式則可將磁碟上整塊內容複製到另一磁碟上，若原磁碟含有已受保護不可複製的檔案，則會印出下列訊息：

```
I/O ERROR
STOPPED AT 32Ø
```

複製主文檔或任何未受保護的檔案，可利用FID 程式。附錄J對FID 完整的指令群有詳細說明。

十一、APPLESOFT的CHAIN功能

欲將PART ONE 程式鏈接(CHAIN)PART TWO 程式，先令CHAIN(BINARY程式)與PART TWO 程式同處一磁碟上，然後在PART ONE 程式後加兩行：

```
PRINT CHR$(4);"BLOAD CHAIN,A52Ø"
CALL  52Ø"PART TWO"
```

附　錄　H

更新 DOS 成爲16段式

更換PROM

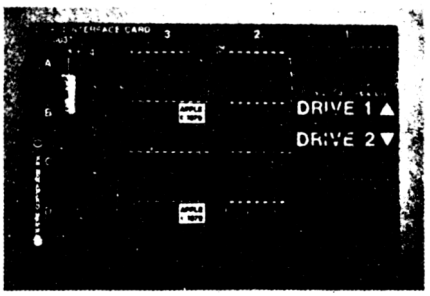
在語言系統上使用DOS

建立—TURNKEY系統

原來RUN 13 段磁碟DOS 的系統（DOS 3.2.1 或更早的版本）可以更換磁碟控製卡上兩個PROM 而更新爲16 段的系統

一、更換PROM

APPLE主板之後端有許多接線槽，其上有一至數片標名“DISKⅡ INTERFACE CARD”的電路卡，將電源關掉後小心抽出電路卡並拉掉其上的帶狀接線。由電路板正面往下數，第二列左邊有一顆標“P6”或“341-ØØ1Ø-XX” 的IC ，在其下電路板印有“PROM P6”。將此IC 換爲“PROM P6A” 。電路板最底下一行亦有一標“P5”或“341-ØØØ9-XX”的IC ，電路板印有“PROM P5” 。將此IC 換爲“PROM P5A”。



新的PROM P6A 及P5A 上應印有“P6A”或“341-ØØ28-XX” ，及“P5A”或“341-ØØ27-XX” 。將PROM P6 以IC 夾小心取出並將PROM P6A 以正確方向插上IC 座，同時亦將P5 換掉。若系統具有兩個或兩個以上磁碟機，各磁碟機機對應的PROM P5 及P6 皆須換新。

將磁碟機接上控制卡，並依下表由上而下順序將控制卡插上接線槽。之後，在磁碟機上標明號數，以利使用。

DISKⅡ 磁碟機位置

裝置順序	溝槽號數	DRIVE號數
第一磁碟機	6	1

裝置順序	溝槽號數	DRIVE號數
第二磁碟機	6	2
第三磁碟機	5	1
第四磁碟機	5	2
第五磁碟機	4	1
第六磁碟機	4	2

檢查各接線及PROM 是否正確穩當，蓋上APPLE蓋子。之後，以MUFFIN將13 段落的磁碟更新爲16 段落的磁碟。

二、在語言系統上使用DOS

插入語言系統卡的APPLEⅡ仍可使用DOS, APPLESOFT及INTEGER BASIC。在語言系統（Pascal 作業系統）中使用語言時，以下列步驟動作16 段DOS：

1. 先將所有有用的檔案存起來，以免遭破壞。
2. 進入PASCAL 作業系統的命令模式（Command Mode）（每行前面會出現COMMAND：）
3. 在第一號磁碟機上，插入DOS 系統主磁碟。
4. 鍵入“H”，以BOOT主磁碟。

主磁碟會將APPLESOFT或INTEGER BASIC（主電路板上沒有的一種）載入語言卡儲存空間。在標準APPLEⅡ機器上，載入APPLESOFT。在APPLE Ⅱ PLUS上載入INTEGER BASIC。此時，顯示幕上會顯示出那一種BASIC被載入。之後，在APPLESOFT或INTEGER BASIC程式的使用或寫入可利用DOS 命令。

未加語言系統（未用POS)的APPLE Ⅱ，其DOS 命令在各BASIC中之使用法可參考相關說明

三、建立—TURNKEY系統

在DOS 3.3 磁碟插入DRIVE 1 並開機後，若HELLO程式所用的BASIC存在機器的ROM中，HELLO 程式就立即被執行。APPLE Ⅱ 可執行以INTEGER BASIC寫的HELLO程式，APPLE Ⅱ PLUS可執行以APPLESOFT寫的HELLO程式。而在含有APPLESOFT卡的APPLE Ⅱ 或含有INTEGER BASIC卡的APPLE Ⅱ PLUS 上則可立即執行任一種BASIC寫的HELLO程式。

若機器中無適當的BASIC執行HELLO程式，則會顯示出“LANGUAGE NOT AVAILABLE”並顯示出已具有的語言。

具有語言系統的APPLE Ⅱ ，會將未存在ROM 中的一種BASIC載入語言卡，使機器亦可執行此BASIC所寫的程式。DOS 3.3 系統主磁碟中有HELLO及APPLESOFT兩個程式，以APPLESOFT語言寫的HELLO程式用來將 INTEGER BASIC載入APPLE Ⅱ PLUS 機器，以INTEGER BASIC 語言寫的APPLESOFT程式用來將APPLESOFT載入APPLE Ⅱ 機器中。視機器種類而定，BOOT之後，兩者之一將機器中ROM 所無的BASIC載入語言卡。之後插入任一DOS3.3磁碟，機器可執行任一BASIC程式。

欲建一個可於任意系統BOOT，並可將適當BASIC載入語言卡的新磁碟，可以以下述方法得之。

1. BOOT DOS 3.3系統主磁碟。
2. LOAD HELLO 並插入一新磁碟，然後
   INIT HELLO
3. 重新插入系統主磁碟
   BRUN MASTER CREATE
   當程式詢及GREETING程式名稱時，鍵入

HELLO　等到程式要求，才再插入空磁碟。

4. 利用FID COPY　下列檔案入新磁碟

APPLESOFT

INTBASIC

FPBASIC

5. 一一將其他必要的檔案放入新磁碟之後，確定新磁碟至少含有以下檔案：

HELLO

APPLESOFT

INTBASIC

FPBASIC

若欲在開機時令系統自動RUN 一使用者的TURNKEY 程式，可將HELLO及APPLESOFT稍作修改。

在修改程式前，先RUN LOADAPA 並使用&S 命令，可見到隱藏的CTRL—D ，如此較易做修改工作。

6. UNLOCK HELLO

7. LOAD HELLO

8. LIST HELLO

```
240  END
250  REM
260  REM --NO CARD OR CAN'T RELOAD
270  REM
280  IF PEEK (768) = 0 THEN END
290  PRINT: PRINT "...LANGUAGE CARD CANNOT BE RELOADED":
     PRINT "   UNTIL THE SYSTEM IS REBOOTED..."
300  END
```

9. 更改240,280,300三行如下：

```
* 240  GOTO 300
  250  REM
  260  REM --NO CARD OR CAN'T RELOAD
  270  REM
* 280  IF PEEK (768) = 0 THEN 300
  290  PRINT: PRINT "...LANGUAGE CARD CANNOT BE RELOADED":
       PRINT "   UNTIL THE SYSTEM IS REBOOTED..."
* 300  PRINT "ctrl-dRUN TURNKEY"
```

其中“ctrl-d” 爲不顯示出的CTRL-D。可改掉，建立含CTRL-D的D$ 取代看不見的CTRL-D:

310 D$=CHR$(4):PRINT D$;“RUN TRUNKEY”

10. SAVE HELLO
11. LOCK HELLO
12. UNLOCK APPLESOFT,LOAD並LIST HELLO

觀察之：

```
lines.
250 GOTO 310
260 REM
270 REM --NO CARD OR CAN'T RELOAD
280 REM
290 IF PEEK (768) = 0 THEN 310
300 PRINT "...LANGUAGE CARD CANNOT BE RELOADED":
    PRINT "   UNTIL THE SYSTEM IS REBOOTED..."
310 PRINT "ctrl-dINT"
```

13. 將程式中290,310 更改如下：

```
250 GOTO 310
260 REM
270 REM --NO CARD OR CAN'T RELOAD
280 REM
* 290 IF PEEK (768) = 0 THEN 310
300 PRINT "...LANGUAGE CARD CANNOT BE RELOADED":
    PRINT "   UNTIL THE SYSTEM IS REBOOTED..."
* 310 PRINT "ctrl-dRUN TURNKEY"
```

第310行亦可改寫如下：

310 D$=“ctrl-d”:PRINT D$;“RUN TURNKEY”

14. SAVE APPLESOFT
15. LOCK APPLESOFT
16. RENAME turnkey 程式TURNKEY 。以下爲以APPLESOFT寫成的TURNKEY 程式例：

```
Applesoft, is below:
100  TEXT : HOME
110  VTAB 3: PRINT "DOS TOOL KIT "; TAB( 30);"30 NOV 1979"
120  VTAB 5: PRINT "(C) COPYRIGHT 1979, APPLE COMPUTER INC."
130  FOR I = 1 TO 1500: NEXT I
140  PRINT "ctrl-dRUN RIBBIT"
```

17. 確定磁碟中含有下列檔案：

```
HELLO
APPLESOFT
INTBASIC
FPBASIC
TURNKEY
```

18. 試驗 turnkey　磁碟。關機後開機，再BOOT　此磁碟，它會載入適當的BASIC並RUN　TURNKEY。

附　錄　I

使用 BASICS 磁碟

以 "BASICS" BOOT磁碟

“BASICS”磁碟用於在16段系統上BOOT 13段磁碟。每當要BOOT 13段磁碟，在開機或鍵入PR#6，IN#6，C600G或6CTRL-P時“BASICS”磁碟都須在磁碟機上。開機後五秒鐘，顯示幕會打出：

INSERT YOUR 13 - SECTOR DISKETTE
AND PRESS RETURN

再插入13段DOS/BASIC磁碟。

已處於DOS上時，可用INT及FP命令自一BASIC轉至另一BASIC語言。在INTEGER BASIC中，催促符號爲>。在APPLESOFT中，催促信號爲]

“BASICS”磁碟作用後，turnkey磁碟仍如前述執行TURNKEY程式。

欲REBOOT磁碟不須先關機，只須插入“BASICS”按PR#6，或IN#6再按RETURN，等到指示信號出現，插入所要BOOT的磁碟。

附　錄　J

使用FID 程式

進入FID程式

給定檔案名稱

FID命令群

"錯誤" 後之處置

FID 程式有兩大功能：

1. 可以 catalog，copy ，delete ，lock 及 unlock DOS 檔案。
2. 只用一個磁碟機便可以 copy 檔案自一磁碟至另一磁碟。

FID 程式應使用於具有 32K 或以上記憶體及一至二個磁碟機的APPLE Ⅱ系統。它是以 APPLESOFT 寫成。FID 的 copy 命令，其目標磁碟須已 INIT。

一、進入FID程式

BOOT系統主磁碟並鍵入

BRUN FID

之後顯示幕上會列出其命令表，只須再鍵入＜＞內數目，就會執行所對應的命令。

二、給定檔案名稱

FID 的命令可以處理一個給定檔名的檔案，或處理以下列方式給定的一群檔案。等號“＝”代表任何可用於檔名的字元。例如檔名“FR=ED”表示所有以“FE”爲首“ED”爲末檔案名稱的檔案。“＝N＝”表示所有含字元“N”檔案名稱的檔案。若只鍵入“＝”，表示磁碟中所有檔案。用等號於檔案名稱的命令，執行時會發出詢問：

DO YOU WANT PROMPTING?

若答以“Y”，則每一符號檔名的檔案處理之前，均會打出檔名並要求使用者再度確定是否處理此檔。應答以“Y”或“N”。使用者欲停止執行剩餘檔案時，可鍵入“Q”。若在“DO YOU WANT PROMPTING”後答以“N”，則所有符合要求的檔均被執行。

三、FID命令羣

大多數FID 命令（ Catalog ，Space ，Delete ，Lock ，Unlock ，及Verify ）作用在上一次使用到或重置回自然值的磁碟機上。Copy 命令可作用在一個或兩個磁碟機上。Quit 命令用不到磁碟機（它只令動作離開FID進入DOS中）。

FID 程式開始執行時並未設定使用那個磁碟機。鍵入的第一個命令若是作用在一個磁碟的，FID 程式會要求使用者輸入此磁碟機的溝槽及DRIVE 號數。其後使用的命令皆使用於此磁碟機上。若進入FID 程式後鍵入的第一個命令爲Copy ，FID 會要求輸入兩磁碟機的溝槽及DRIVE 號數（一爲來源磁碟，一爲目標磁碟）。每當Copy 命令與作用在一磁碟的命令交互使用時，使用者須設定一或二個磁碟機。使用者可隨時利用Reset 命令重定磁碟。

四、目錄——Catalog

鍵入“２”，會顯示出已定磁碟上的檔案目錄。

五、重置溝槽及DRIVE——Reset

鍵入“７”，會取消已定的磁碟溝槽及DRIVE 號數。使用下一個命令時須設新的磁碟溝槽及DRIVE 號數。（本Reset命令與鍵盤上之RESET 鍵無關。

六、鎖制檔案——Lock

鍵入“５”，等到催促信號出現再鍵入檔名，以鎖制此檔。之後，應將存有此檔的磁碟插入磁碟機，並再度確定是否鎖制檔案或ESC ape 掉此命令，若決定鎖制此檔，磁碟上的此檔即被鎖制。此被鎖制檔案在Catalog 上其檔名前會有星號標出。若磁碟上無指定檔案，則顯示出：

NO FILES SELECTED

之後可以隨便按一鍵（除了RESET ，SHIFT ，CTRL ，及ESC ）以回復程式 。

七、解鎖檔案——Unlock

鍵入“ 4 ”，等到催促信號出現後鍵入檔名。將存此檔的磁碟插入磁碟機，並再度確定是否解制此檔或ESCape掉此命令，若決定解鎖此檔，磁碟上的此檔即被解鎖。在Catalog上檔名前星號不再標出 。若磁碟上無指定檔案，則顯示出：

NO FILES SELECTED

之後可以隨便按一鍵（除了RESET ，SHIFT ，CTRL ，及ESC）以回復程式 。

八、查證——Verify

鍵入“ 8 ”，回答催促信號，當查證完指定檔案後，會打出檔名及“ DONE ”於顯示幕。若指定的檔案爲不可讀，則會顯示出Ｉ／Ｏ錯誤信號 。

本命令與DOS 的VERIFY 命令具相同功能 。

九、複製檔案——Copy

鍵入“ 1 ”，之後再鍵入來源磁碟機及目標磁碟機的溝槽及DRIVE 號數 。

A、複製檔案於兩磁碟機間

使用者鍵入檔案名稱後（可用通用符號“ = ”表示一群檔案），應插入適當的磁碟，再按下“ ESC ”（取消Copy 工作）或除了“ RESET ”，“ SHIFT ”，“ CTRL ”，“ ESC ”之外的任一鍵（執行Copy 工作）。

若來源磁碟存有指定檔案，目標磁碟也有足夠空間存入，執行之後APPLE 會通知使用者Copy 動作完成。否則，亦通

知DOS 錯誤訊息。

若目標磁碟已具有同名檔案，則會顯示出下列訊息：

```
FILE
ALREADY EXISTS.
 TYPE IN NEW FILE NAME FOR THE COPY OR
<RETURN> TO REPLACE PRESENT FILE OR
<CTRL-C><RETURN> TO CANCEL COPY
:
```

若鍵入新檔名，複製過來的檔案便以此名存入目標磁碟。若按“ RETURN ”鍵，舊檔案亦未鎖制，此時複製過來的新檔案就取代了舊檔案。若按“ RETURN ”鍵，而舊檔案已鎖制了，則顯示如下：

```
FILE LOCKED.
DO YOU WISH TO REPLACE IT ANYWAY?
```

若鍵入“ Y ”，則新檔案取代舊檔案，若鍵入“ N ”，則再度顯示前述一群訊息。之後鍵入“ CTRL-C ”及“ RETURN ”取消Copy 功能，若未曾複製檔案，顯示出：

NO FILES SELECTED

若曾複製檔案，則顯示出“ DONE ”，並顯示出：

PRESS ANY KEY TO CONTINUE

按下一鍵繼續執行ＦID 程式。

B、在單一磁碟機上複製檔案

若鍵入“ 1 ”之後，給定相同的來源磁碟機及目標磁碟機溝槽及DRIVE 號數，程式會適時要求使用者插入來源磁碟及目標磁碟。如此便可在同一磁碟機上Copy 兩磁碟。

九、刪除檔案—— Delete

鍵入“ 6 ”，程式會詢問“ FILENAME？”，鍵入檔名

後，程式開始執行除檔工作，除檔完畢後會顯示出“＜DONE＞”。若找不到檔案，則顯示“NO FILES SELECTED”。若找到的檔案已鎖制，則顯示“FILE LOCKED”在得到催促信號“PRESS ANY KEY TO CONTINUE ”後，按下任一鍵（除了“RETURN ”，“SHIFT ”，“CTRL” ，“ESC ”）回到FID 程式。

十、查磁碟空間——Space

鍵入“3”，磁碟上已使用的段落數目及仍未使用的段落數目會顯示出來。

十一、中止執行——Quit

鍵入“9”，跳出FID 程式回到原來的狀態（BASIC）。

十二、“錯誤”後之處置

程式執行時若發生錯誤，會打出錯誤訊息，並中止執行。當Copy 程式因此遭中止執行時，使用者應查看目標磁碟的目錄，將其中未完成的同名或給定新名的假檔案除掉。

錯誤訊息包括：DISK FULL ，DISK WRITE PROTECTED ，FILE LOCKED，及I／O ERROR 。遇到錯誤訊息後，只須按下任一鍵便可回復到FID 程式。

若DOS 錯誤訊息非上述幾個之一，則會打出附帶的錯誤碼。這只發生在企圖破壞FID 或DOS ，或複製不良的檔案之時。若發生了此事，使用者只能re boot DOS並再度“BRUN FID ”。

附　錄　K

使用 MUFFIN 程式

使用MUFFIN程式

利用通用字元 "二"

DOS 3.3爲16 段磁碟的DOS ，早於DOS3.3 的版本皆爲13 段磁碟的DOS 。關於磁碟段落問題，可參考附錄C的第一段部分。

因爲在13 段磁碟上資料安排方式的不同，不能在16 段DOS 系統上使用13 段磁碟。若在16 段DOS 系統使用13 段磁碟，系統會打出錯誤訊息：

UNABLE TO READ/WRITE

一、使用MUFFIN程式

MUFFIN 程式可將13 段磁碟中的資料重新安排並寫入另一磁碟，使其轉爲正常的16 段磁碟資料。此目標磁碟在寫入之前應先在16 段DOS 系統上 INIT。在資料自來源磁碟（ 13 段）轉至目標磁碟後，來源磁碟上的13 段DOS 版本仍完整保存住。

MUFFIN 程式可對一個或兩個磁碟機作用。以下例子說明MUFFIN 程式在一個磁碟機上自13 段磁碟轉換檔案至16 段磁碟的方法。假設此磁碟機接在第六溝槽磁碟控制卡上。

1. 在磁碟機上插入系統主磁碟，鍵入

   BRUN MUFFIN

2. 下列訊息出現後，鍵入“ 1 ”，表示欲轉換檔案。

```
*****************************************
*  APPLE ][ DOS 3.2 TO 3.3 CONVERTER    *
*                                       *
*            MUFFIN VERSION D           *
*                                       *
*  COPYRIGHT 1979 APPLE COMPUTER INC.   *
*****************************************
CHOOSE ONE OF THE FOLLOWING OPTIONS
        <1>   CONVERT FILES
        <2>   QUIT
WHICH WOULD YOU LIKE?
```

3. 顯示幕出現“ SOURCE SLOT？”時，鍵入“ 6 ”（磁碟控制卡所在溝槽號數）。顯示幕再出現“ DRIVE ？”

，鍵入“ 1 ”。。

4. 顯示幕出現“ DESTINATION SLOT ？”時，鍵入“ 6 ”。顯示幕出現“ DRIVE ？”，鍵入“ 1 ”。

5. 之後，程式詢及檔案名稱：

   FILENAME?

   鍵入“ = ”等號表示轉換來源磁碟（ 13 段）上所有的檔案。

6. 在轉換之前，程式會詢及：

   DO YOU WANT PROMTING ?

   鍵入“ N ”表示不須要。

7. 當出現以下訊息時，將系統主磁碟抽出，插入13 段來源磁碟並按“ RETURN ”鍵。

```
INSERT DISKS THEN PRESS <ESC> TO RETURN TO
MAIN MENU OR ANY OTHER KEY TO BEGIN
```

8. 出現以下訊息時，只要按“ RETURN ”鍵即可。

   INSERT SOURCE DISK AND PRESS A KEY

9. MUFFIN 程式找出13 段來源磁碟上第一個檔案，並將檔案名稱打出。之後，使用者換上已 INIT的16 段磁碟，並按“ RETURN ”鍵。一會而後顯示幕出現“ DONE ”信號表示完成轉換此檔，並指示使用者再換上 13 段磁碟以便繼續做下一個檔案。重複這過程直到所有檔案轉換完畢爲止。

註：轉換一個極大的檔案，可能須輪番換上來源磁碟與目標磁數次。

在多於一個磁碟機上轉換檔案時，只要注意在第 3 及第 4 步驟

上鍵入適當的溝槽及DRIVE 號數，執行前正確地將來源磁碟與目標磁碟插入所定的個別磁碟機上。

在第 6 步驟，得到下列訊息後，使用者可以改變主意按下“ ESC ”鍵，使程式停止，回頭重新考慮。

若在目標磁碟原有同名的檔案，則顯示出：

```
FILE [filename]
ALREADY EXISTS.
TYPE IN A NEW FILE NAME FOR THE COPY
OR <RETURN> TO REPLACE EXISTING FILE
OR <CTRL-C><RETURN> TO CANCEL COPY:
```

此時可以鍵入一新檔名作爲轉換檔案的檔案名稱，或按“ RETURN ”鍵以轉換檔案取代目標磁碟的原檔案，或按下“ CTRL-C ”及“ RETURN ”停止轉換。

二、利用通用字元“＝”（Wildcard charcater）

通用字元“＝”可代表檔名中任一未寫出的字元或字元群。如：“＝”表示磁碟中所有的檔案。回答“ FILENAME？”詢問時，答以“ FI=LE ”表示欲轉換 13 段磁碟中所有以 FI 爲開始，LE 爲結束檔名的檔案。“＝＊＝”則表示含星號檔案名稱的所有檔案。

使用通用字元後會出現“ DO YOU WANT PROMPTING？”若答以“ Y ”，程式每在 13 段磁碟找到一個檔案就停下等使用者確定是否轉換它（ Y，N，Q三種選擇，各表示欲轉換，不轉換，及停止程式）。若在“ DO YOU WANT PROMPTING ？”後答以“ N ”，表示毫不考慮地將所有符合的檔案轉換過來。

PUBLISHED & PRINTED IN HONG KONG H.K

APPLE DOS 作業系統使用手冊

蔡毓琛
魏易休
譯

香港科技出版社